

CC

CC

```

376                                     .SBTTL $RDBQP - Purge Buffer Wait Queue
377
378 *--$RDBQP- Purge Buffer Wait Queue
379
380 This routine is called to cancel all requests in the Receive Data
381 Buffer Wait Queue for a particular system line number.
382
383 Inputs:
384 R4 contains the system line number
385
386 Outputs:
387 None.
388
389
390 000500 $RDBQP::SAVRG <R3,R4> ; Save some registers
391
392 .IF DF M$$PRO
393 CALL $MPSAV ; Bypass the cache
394 .ENDC
395
396 000504 006304 ASL R4 ; Convert SLN into word offset
397 000506 066704 ADD $SLTMA,R4 ; Point into System Line Index Table
398 000512 013403 MOV @R4,R3 ; Get System Line Table flags word
399 000514 042703 BIC #^C<LF.BWT>,R3 ; Isolate wait request count
400 000520 040354 BIC R3,@R4 ; Clear wait request count in SLT
401 000522 160367 SUB R3,$RDOCT ; Reduce global wait count by same value
402
403 .IF DF M$$PRO
404 CALL @C(SP)+ ; Restore state of the cache
405 .ENDC
406
407 000526 RESRG <R4,R3> ; Restore registers
408 000532 RETURN

```

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
CALL	6-100	8-165	8-173	8-176	9-227	10-249	11-287	11-296	12-328	12-330
CCBDF\$	#5-60	5-62								
ENABL\$	#5-59	6-106	8-183	10-255	12-336					
INHIB\$	#5-59	6-94	8-159	9-211	10-243	11-281	12-318			
MAP	#5-59	18-582	19-619							
RESMAP	#5-59	8-185	19-622							
RESRG	#5-59	8-184	12-337	13-373	14-407					
RETURN	6-107	7-135	8-186	10-256	12-338	13-374	14-408	15-480	16-535	18-592
	18-598	19-625								
SAVMAP	#5-59	8-156	19-618							
SAVRG	#5-59	8-157	9-210	11-279	12-317	13-358	14-390			
SLTDF\$	#5-60	5-63								

```

376                                     .SBTTL $RDBQP - Pu ge Buffer Wait Queue
377
378 :+
379 :*- $RDBQP- Purge Buffer Wait Queue
380 :
381 :   This routine is called to cancel all requests in the Receive Data
382 :   Buffer Wait Queue for a particular system line number.
383 :
384 : Inputs:
385 :   R4 contains the system line number
386 :
387 : Outputs:
388 :   None.
389 :-
390 000500 $RDBQP::SAVRG <R3,R4> ; Save some registers
391
392 .IF DF M$$PRO
393 CALL $MPSAV ; Bypass the cache
394 .ENDC
395
396 000504 006304 ASL R4 ; Convert SLN into word offset
397 000506 066704 ADD $SLTMA,R4 ; Point into System Line Index Table
398 000512 013403 MOV @R4+,R3 ; Get System Line Table flags word
399 000514 042703 BIC #^C<LF.BWT>,R3 ; Isolate wait request count
400 000520 040354 BIC R3,@-R4 ; Clear wait request count in SLT
401 000522 160367 SUB R3,$RDPCT ; Reduce global wait count by same value
402
403 .IF DF M$$PRO
404 CALL @SP+ ; Restore state of the cache
405 .ENDC
406
407 000526 RESRG <R4,R3> ; Restore registers
408 000532 RETURN

```


MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
CALL	7-100	9-165	9-173	9-176	10-227	11-249	12-287	12-296	13-328	13-330
	16-454	17-525								
CCBDF\$	#6-60	6-62								
ENABL\$	#6-59	7-106	9-183	11-255	13-336					
INHTE\$	#6-59	7-94	9-159	10-211	11-243	12-281	13-318			
	#6-59	19-582	20-619							
RESMAP	#6-59	9-185	20-622							
RESRG	#6-59	9-184	13-337							
RETURN	7-107	8-135	9-186	14-373	15-407	16-456	17-526			
	19-592	19-538	20-625	11-256	13-338	14-374	15-408	16-480	17-528	17-535
SAVMAP	#6-59	9-156	20-618							
SAVRG	#6-5	9-157	10-210	12-279	13-317	14-358	15-390	16-447	17-516	
SLTDF\$	#6-60	6-63								

```

376                                     .SBTTL DDMDSP - DISPATCH TO DDM LEVEL
377
378                                     ;+
379                                     ;*-DDMDSP-DISPATCH TO DDM LEVEL
380                                     ;
381                                     ; THIS SUBROUTINE IS CALLED TO DISPATCH DOWN TO THE DDM LEVEL. PROCESS
382                                     ; DISPATCH WILL OCCUR AT THE CURRENT PRIORITY OR AT THE PROCESS PRIORITY
383                                     ; DEPENDENT ON A PDV FLAG.
384                                     ;
385                                     ; INPUTS:
386                                     ;
387                                     ; R2 = SYSTEM LINE NUMBER
388                                     ;
389                                     ; REGISTERS MODIFIED:
390                                     ;
391                                     ; R5
392                                     ;-
393
394 DDMDSP: CALL $STDD1 ; SET UP DDM PDV INDEX & LINE TABLE ADDRESS
395          CLC ; SAVE CURRENT PRIORITY
396          MFPS -(SP) ; WITH C-BIT CLEAR
397          SAVRG <R0> ; GET A FREE REGISTER
398          MOV R2,R0 ; COPY DDM PDV INDEX
399          ADD $PDVTA,R0 ; POINT INTO PDV INDEX TABLE
400          MOV (R0),R0 ; GET POINTER TO PDV
401          BIT #ZF.DVP,Z.FLG(R0)
402          BEQ 10$ ; IF EQ, RUN DDM AT CURRENT PRIORITY
403
404          MTPS #PR7 ; RAISE PROCESSOR PRIORITY
405
406          .IF DF L$$$I1
407
408          MFPS -(SP) ;;; GET CURRENT PRIORITY
409          BICB Z.SCH(R0),(SP) ;;; SET UP PROCESS PRIORITY
410          MTPS (SP)+ ;;; SET UP PROCESSOR PRIORITY
411
412          .IFF
413
414          BICB Z.SCH(R0),PS ;;; DROP TO PROCESS PRIORITY
415
416          .ENDC
417
418          10$: RESRG <R0> ; RESTORE REGISTER
419          CALL $PDDSP ; DISPATCH TO THE DDM
420          ADC (SP) ; UPDATE C-BIT IN SAVED PS
421          MTPS (SP)+ ; RESTORE PROCESSOR PRIORITY
422          RETURN

```

7-	117	\$DD??? - DLC TO DDM REQUESTS
9-	196	DDCM1 - COMMON PROCESS FOR SPECIAL NON-CCB FUNCTIONS
10-	236	DDCM2 - COMMON PROCESS FOR MODEM CONTROL FUNCTIONS
11-	317	DDCM3 - COMMON PROCESS FOR FUNCTIONS WITH CCB
11-	318	DDCM4 - COMMON PROCESS FOR FUNCTIONS WITH CCB
12-	376	DDMDSP - DISPATCH TO DDM LEVEL
13-	424	\$DDAST - ASYNCHRONOUS COMPLETION TO DLC LEVEL
14-	463	\$DDXMP - TRANSMIT COMPLETE TO DLC LEVEL
14-	464	\$DDRCP - RECEIVE COMPLETE TO DLC LEVEL
14-	465	\$DDCCP - CONTROL COMPLETE TO DLC LEVEL
14-	466	\$DDKCP - KILL COMPLETE TO DLC LEVEL
15-	533	\$STDUM - SET DDM PDV INDEX AND LINE TABLE ADDRESS

\$DDXMP - TRANSMIT COMPLETE TO DLC LEVEL

```

463 .SBTTL $DDXMP - TRANSMIT COMPLETE TO DLC LEVEL
464 .SBTTL $DDRCR - RECEIVE COMPLETE TO DLC LEVEL
465 .SBTTL $DDCCP - CONTROL COMPLETE TO DLC LEVEL
466 .SBTTL $DDKCP - KILL COMPLETE TO DLC LEVEL
467
468 ***$DDXMP-TRANSMIT COMPLETE TO DATA LINK CONTROL
469 ***$DDRCR-RECEIVE COMPLETE
470 ***$DDCCP-CONTROL COMPLETE
471 ***$DDKCP-KILL COMPLETE
472 ***$LLCSP-SPECIAL ENTRY POINT FOR X25
473
474 THIS SUBROUTINE IS CALLED BY DEVICE DRIVERS TO QUEUE
475 COMPLETION NOTIFICATIONS TO DATA LINK CONTROL MODULES.
476
477 THE $LLCSP ENTRY POINT IS PROVIDED FOR USE BY THE X25 DATA LINK MAPPING.
478
479 INPUTS:
480
481 R3 = OPERATION COMPLETION STATUS
482 R4 = ADDRESS OF CCB TO QUEUE
483 THE CCB CONTAINS A VALID LINE NUMBER
484
485
486 OUTPUTS:
487
488 THE CCB IS QUEUED TO A DLC LIST BASED
489 ON THE SYSTEM LINE NUMBER PARAMETER IN THE CCB WITH A
490 TRANSMIT OR RECEIVE COMPLETE FUNCTION CODE.
491
492 REGISTERS MODIFIED:
493
494 XXX
495
496
497 000430 112764 000012 000010 $DDXMP::MOVB #FC.XCP,C.FNC(R4) ; SET TRANSMIT COMPLETE FUNCTION CODE
498 000436 000420 BR 20$ ; JOIN COMMON CODE
499
500 000440 112764 000014 000010 $DDRCR::MOVB #FC.RCP,C.FNC(R4) ; SET RECEIVE COMPLETE FUNCTION CODE
501 000446 000414 BR 20$ ; JOIN COMMON CODE
502
503 000450 112764 000016 000010 $DDKCP::MOVB #FC.KCP,C.FNC(R4) ; SET KILL COMPLETE FUNCTION CODE
504 000456 000410 BR 20$ ; JOIN COMMON CODE
505
506 000460 005764 000006 $DDCCP::TST C.LIN(R4) ; SPECIAL MODEM CONTROL COMPLETION?
507 000464 100002 BPL 10$ ; NO
508 000466 CALLR $CCBRT ; YES - RETURN CCB TO POOL
509 ; MODEM CONTROLLER WILL POST COMPLETION
510 000472 112764 000020 000010 10$: MOVB #FC.CCP,C.FNC(P4) ; SET CONTROL COMPLETE FUNCTION CODE
511
512 $LLCSP::
513 $DDSPC::
514 000500 010364 000012 20$: MOV R3,C.STS(R4) ; SET COMPLETION STATUS
515
516 .IF DF M$$PRO
517
518 MOVB C.LIN(R4),R3 ; GET SYSTEM LINE #
519 ASL R3 ; FORM WORD OFFSET

```

```
58  
59  
60  
61  
62  
63  
64  
65  
66 000000  
67 000000  
68 000000  
69  
:  
: MACRO LIBRARY CALLS  
:  
: .MCALL INHIB$,ENABL$,SAVRG,RESRG  
: .MCALL CCBDF$,PDVDF$,SLTDF$  
: .MCALL CALLR  
: AVOID SYSTEM DEPENDENCY  
: DEFINE THE CCB OFFSETS  
: DEFINE THE PDV OFFSETS  
: DEFINE THE SLT OFFSETS
```

CEDDMN MACRO V05.03b Friday 28-Jun-85 18:18 Page 15
 \$STDDM - SET DDM PDV INDEX AND LINE TABLE ADDRESS

```

533 .SBTTL $STDDM - SET DDM PDV INDEX AND LINE TABLE ADDRESS
534
535
536 *+
537 **-$STDDM-SET DDM PDV INDEX AND LINE TABLE ADDRESS
538 **-$STDD1-(ALTERNATE ENTRY)
539
540 THIS SUBROUTINE IS CALLED TO SET UP A DEVICE DRIVER PDV INDEX AND
541 LINE TABLE ADDRESS BASED ON A SYSTEM LINE NUMBER.
542
543 INPUTS:
544     R2 = SYSTEM LINE NUMBER (ALTERNATE ENTRY ONLY)
545     R4 = CCB ADDRESS WITH A VALID SYSTEM LINE NUMBER IN
546         C.LIN (MAIN ENTRY ONLY)
547
548 OUTPUTS:
549     R2 = PDV INDEX
550     R5 = ADDRESS OF DEVICE LINE TABLE
551
552 REGISTERS MODIFIED:
553     R5
554
555 -
556
557
558 000000 $STDDM::IF NDF N$$1LN
559
560 000000 116*02 000006      MOVB     C.LIN(R4),R2      ; EXTRACT SYSTEM LINE NUMBER
561
562 .ENDC
563
564 000004 $STDD1::IF DF N$$1LN
565
566      MOV     @$$SLTMA,R2      ; GET ADDRESS OF SYSTEM LINE ENTRY
567
568 .IFF
569
570 000004 006302      ASL     R2      ; FORM WORD INDEX
571 000006 066702      ADD     $$SLTMA,R2      ; POINT INTO SYSTEM LINE INDEX TABLE
572 000012 011202      MOV     (R2),R2      ; GET ADDRESS OF SYSTEM LINE ENTRY
573
574 .ENDC
575
576 000014 016205 000004      MOV     L.DDS(R2),R5      ; GET DEVICE LINE TABLE ADDRESS
577 000020 116202 000002      MOVB     L.DDM(R2),R2      ; GET DEVICE DRIVER PDV INDEX (WORD INDEX)
578 000024      RETURN
579
580
581 000001 .END

```

```
204 CLR C.URM(R4) ; ALLOW PROCESS TO RUN ON ANY PROCESSOR
205
206 .ENDC
207
208 000164 CALL $PDQUE ; QUEUE CCB AND SCHEDULE PROCESS
209 000170 012602 MOV (SP)+,R2 ; RESTORE R2
210 000172 000241 CLC ; CLEAR C-BIT
211 000174 30$: RETURN ; RETURN TO CALLER
212
213 .DSABL LSB
214
215 000001 .END
```

```

128 .SBTTL $LLCRS - LLC TO LLC REQUEST QUEUING SUBROUTINE
129
130 *-- $LLCRS - LLC TO LLC REQUEST QUEUING SUBROUTINE
131 *-- $DLCRS - DLC TO LLC REQUEST QUEUING SUBROUTINE FOR X25
132 *-- $NMCRS - LLC TO DLC REQUEST QUEUING SUBROUTINE FOR NETWORK MANAGEMENT
133
134 THIS ROUTINE IS CALLED BY A LLC PROCESS TO QUEUE REQUESTS TO
135 ANOTHER LLC PROCESS. NOTE THAT THE CALLERS CONTEXT IS SAVED AS
136 IF AN INTERRUPT HAD OCCURRED.
137
138 THE $DLCRS ENTRY POINT IS SUPPLIED FOR USE BY THE X25 DATA LINK MAPPING.
139
140 THE $NMCRS ENTRY POINT IS SUPPLIED FOR USE BY NETWORK MANAGEMENT SO THAT
141 IT CAN FILL IN THE UNIBUS RUN MASK BEFORE ISSUING A REQUEST TO A DLC
142 FOR LINE/CIRCUIT COUNTERS ETC.
143
144 INPUTS:
145 R4= ADDRESS OF CCB (OR FIRST IN CHAIN)
146 THE CCB MUST CONTAIN A VALID:
147 C.FNC & C.MOD - REQUEST FUNCTION CODE
148 C.STA - DESTINATION LLC'S PDV INDEX
149 C.LIN - OPTION CALLING PARAMETER
150
151 OUTPUTS:
152 THE REQUEST IS QUEUED AND THE APPROPRIATE LEVEL IS SCHEDULED
153 NOTE: A LLC PROCESS CAN DETERMINE IF A COMPLETION HAS COME FROM
154 A DLC PROCESS OR ANOTHER LLC BY EXAMINING THE MSB
155 OF THE C.LIN/C.STA CELL IN THE CCB.
156 MSB=0 THE REQUEST CAME FROM A DLC AND THE BYTE C.LIN
157 CONTAINS THE LLC'S CHANNEL NUMBER FOR THE LOGICAL
158 LINE THAT HAD THE COMPLETION.
159 MSB=1 THE REQUEST CAME FROM A LLC AND THE BYTE C.LIN
160 CONTAINS AN OPTIONAL CALLING PARAMETER FROM THE
161 REQUESTING LLC.
162
163 REGISTERS MODIFIED:
164 NONE
165
166
167 000074 $DLCRS::
168 000074 $LLCRS:: IF DF M$PRO
169
170 CLR C.URM(R4) ; ALLOW PROCESS TO RUN ON ANY PROCESSOR
171
172 .ENDC
173
174 000074 $NMCRS:: SAVRG <R3,R4> ; SAVE R3 AND R4
175 000100 052764 100000 000006 BIS #100000,C.LIN(R4) ; FLAG DESTINATION PROCESS IS LLC
176 000106 152764 000200 000003 BISB #200,C.BID(R4) ; MARK CCB AS COMING FROM ANOTHER LLC
177
178 000114 CALL $PDQU1 ; QUEUE REQUEST CCB AND SCHEDULE PROCESS
179 000120 RESRG <R4,R3> ; RESTORE REGISTERS
180 000124 RETURN ; AND RETURN TO CALLING LLC

```


CELOG - COMM/EXEC EVENT LOGGING MACRO V05.03b Friday 28-Jun-85 18:19 Page 5-2

Symbol table

A\$\$CHK= 000000	I\$\$RAR= 000000	N\$\$ACC= 000001	T\$\$KMG= 000000	ZF.PSE= 002000
A\$\$CPS= 000000	I\$\$RDN= 000000	N\$\$BUF= 000001	T\$\$MIN= 000000	ZF.SLT= 010000
A\$\$PRI= 000000	K\$\$CNT= 177546	N\$\$LDV= 000001	V\$\$CTR= 001000	ZF.TIM= 000200
A\$\$TRP= 000000	K\$\$CSR= 177546	N\$\$MCP= 000001	X\$\$DBT= 000000	ZF.X3P= 000000
C\$\$CKP= 000000	K\$\$L= 000000	N\$\$MLL= 000001	ZF.COU= 001000	ZS.ASN= 100000
C\$\$DRE= 000400	K\$\$P= 000074	N\$\$MOV= 000010	ZF.DDM= 000001	ZS.BSY= 140000
C\$\$RSH= 177546	LDF= 000000	N\$\$ACT= 000001	ZF.DIA= 004000	Z.AVL= 000014
D\$\$BUG= 177514	L\$\$ASG= 000000	N\$\$PEM= 000001	ZF.DLC= 000002	Z.DAT= 000016
D\$\$ISK= 000000	L\$\$DRV= 000000	P\$\$P45= 000000	ZF.DVP= 100000	Z.DSP= 000000
D\$\$L11= 000001	L\$\$P11= 000001	P\$\$WRD= 000000	ZF.INI= 040000	Z.FLG= 000010
D\$\$YNC= 000000	L\$\$11R= 000000	Q\$\$OPT= 000010	ZF.KMX= 000020	Z.LEN= 000016
D\$\$YNM= 000000	M\$\$CRE= 000124	R\$\$DER= 000000	ZF.LLC= 000004	Z.LLN= 000006
E\$\$XPR= 000000	M\$\$CRV= 000000	R\$\$K11= 000001	ZF.LMC= 000100	Z.MAP= 000020
F\$\$LVL= 000001	M\$\$FCS= 000000	R\$\$SND= 000000	ZF.MAN= 020000	Z.NAM= 000004
G\$\$TIP= 000000	M\$\$MGE= 000000	R\$\$11M= 000000	ZF.MFL= 000010	Z.PCB= 000012
G\$\$TSS= 000000	M\$\$NET= 000000	S\$\$WRG= 000000	ZF.MTM= 000400	Z.SCH= 000007
G\$\$TTK= 000000	M\$\$OVR= 000000	S\$\$YSZ= 007600	ZF.MUX= 000040	\$CELOG 000000RG
G\$\$WRD= 000000				

. ABS. 000020 000 (RW,I,GBL,ABS,OVR)
 000002 001 (RW,I,LCL,REL,CON)

Errors detected: 0

*** Assembler statistics

Work file reads: 0
 Work file writes: 0
 Size of work file: 10159 words (40 Pages)
 Size of core pool: 14440 words (55 Pages)
 Operating system: RSX-1 M/PLUS

Elapsed time: 00:00:06.39

SY:CELOG.V2,[130,134]CELOG/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]CELOG

.TITLE CESCH
.IDENT /V05.00/

..COPYRIGHT (C) 1978,1979,1980,1982,1983, 1984, 1985 BY
..DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

..THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
..ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
..INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
..COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
..OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
..TRANSFERRED.

..THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
..AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
..CORPORATION.

..DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
..SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

..MODULE DESCRIPTION

..CEX PROCESS SCHEDULING ROUTINES

..DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

..IDENT HISTORY:

- ..1.00 10-FEB-78
..VERSION 2.0 RELEASE
- ..2.00 14-DEC-79
..DECNET-11M/S V3.0
..DECNET-11M-PLUS V1.0
- ..3.00 16-APR-82
..DECNET-11M V3.1
..DECNET-11M-PLUS V1.1
- ..4.00 07-NOV-83
..DECNET-11M V4.0
..DECNET-11M-PLUS V2.0
- ..5.00 22-JUL-85
..DECnet-11m/s V4.2
..DECnet-11M-Plus V3.0
..DECnet-Micro/Rsx V1.0

CESCH CREATED BY MACRO ON 28-JUN-85 AT 18:20

PAGE 1 B 15

SYMBOL CROSS REFERENCE

CREF 04.00

SYMBOL VALUE REFERENCES

CF.DDM	=	000002	G	#4-65																
CF.DYN	=	000004	G	#4-65																
CF.EIS	=	000010	G	#4-65																
CF.FRK	=	100000	G	#4-65																
CF.LOG	=	000020	G	#4-65																
CF.MDM	=	000001	G	#4-65																
CF.TIM	=	000400	G	#4-65																
CS.LST	=	040000		9-311	9-315															
C.LIN	=	000006		*9-306																
C.STS	=	000012		*9-311	*9-315															
I\$SAS	=	*****		4-64																
KISAR5	=	*****	GX	*5-132	*6-177	8-247	*8-255	*8-269												
KISAR6	=	*****	GX	8-246	8-270															
K\$SDAS	=	*****		5-112	6-172	8-249	8-264													
L\$S11	=	*****		5-93	7-208	9-317	9-348													
M\$SPRO	=	*****		5-100	5-106	9-319	9-344													
PR7	=	*****		5-110	7-203	9-317														
PS	=	*****	GX	*5-97	*5-110	*7-203	*7-214	9-317	*9-317	*9-348										
R\$SMPL	=	*****		4-67	9-331	10-361	10-370													
R\$S11D	=	*****		4-64																
R\$S11M	=	000000		4-64																
R\$S11S	=	*****		4-64																
X\$MCB	=	*****		4-64	4-64															
ZF.COU	=	001000		#4-64																
ZF.DDM	=	000001		#4-64																
ZF.DIA	=	004000		#4-64																
ZF.DLC	=	000002		#4-64																
ZF.DVP	=	100000		#4-64																
ZF.INI	=	040000		#4-64																
ZF.KMX	=	000020		#4-64																
ZF.LLC	=	000004		#4-64																
ZF.LMC	=	000100		#4-64																
ZF.MAN	=	020000		#4-64																
ZF.MFL	=	000010		#4-64																
ZF.MTM	=	000.00		#4-64																
ZF.MUX	=	000040		#4-64																
ZF.PSE	=	002000		#4-64																
ZF.SLI	=	010000		#4-64																
ZF.TIM	=	000200		#4-64																
ZF.X3P	=	000000		#4-64																
ZS.ASN	=	100000		#4-64																
ZS.BSY	=	140000		#4-64																
Z.AVL	=	000014		#4-64																
Z.DAT	=	000016		#4-64																
Z.DSP	=	000000		#4-64	4-64															
Z.FLG	=	000010		#4-64																
Z.LEN	=	000016		#4-64																
Z.LLN	=	000006		#4-64																
Z.MAP	=	000020		#4-64																
Z.NAM	=	000004		#4-64																
Z.PCB	=	000012		#4-64																
Z.SCH	=	000007		#4-64	7-214															

CESUB MACRO V05.03b Friday 28-Jun-85 18:20 Page 10
\$CALLX - MAPPED SUBROUTINE CALL

```

310 .SBTTL $CALLX - MAPPED SUBROUTINE CALL
311
312 *--$CALLX-MAPPED SUBROUTINE CALL TO ANOTHER PROCESS
313
314 CALLING SEQUENCE:
315 JSR R5,$CALLX
316 .WORD <ADDRESS OF SUBROUTINE TO CALL>
317 .RAD50 <PROCESS NAME>
318
319 NOTE THAT THIS SUBROUTINE ASSUMES THAT THE NAMED PROCESS EXISTS IN THE
320 SYSTEM AND DOES NOT CHECK IF THE PROCESS IS LOADED.
321
322
323 000006 .IIF NDF K$$DAS OFS=6 ; STACK OFFSET FOR R5
324 .IIF DF K$$DAS OFS=10 ; STACK OFFSET FOR R5
325
326 000316 016746 000000G $CALLX::MOV $CMPDV,-(SP) ; SAVE CURRENT PDV INDEX
327 000322 016746 000000G MOV KISAR5,-(SP) ; AND CURRENT PROCESS MAPPING
328
329 .IF DF K$$DAS
330 MOV KINAR5,-(SP) ; SAVE INSTRUCTION MAPPING
331 .ENDC ; DF K$$DAS
332
333 000326 012546 MOV (R5)+,-(SP) ; SAVE DESTINATION SUBROUTINE ADDRESS
334 000330 SAVRG <R1,R2> ; GET 2 FREE REGISTERS
335 000334 016701 000000G MOV $PDVTA,R1 ; POINT TO PDV INDEX TABLE
336 000340 012102 10$: MOV (R1)+,R2 ; GET POINTER TO NEXT PDV
337 000342 001776 BEQ 10$ ; IGNORE ZERO ENTRIES
338 000344 026215 000004 CMP Z,NAM,R2),(R5) ; DO WE HAVE A MATCH?
339 000350 001373 BNE 10$ ; IF NE, NO
340 000352 005725 TST (R5)+ ; SKIP OVER PROCESS NAME
341 000354 011267 000000G MOV (R2),KISAR5 ; MAP TO DESTINATION PROCESS
342
343 .IF DF K$$DAS
344 MOV (R2),KINAR5 ; ALSO MAP INSTRUCTION SPACE
345 .ENDC ; DF K$$DAS
346
347 000360 005741 TST -(R1) ; BACK UP PDV INDEX TABLE POINTER
348 000362 166701 000000G SUB $PDVTA,R1 ; COMPUTE PDV INDEX
349 000366 010167 000000G MOV R1,$CMPDV ; AND SAVE IT
350 000372 016602 000012 MOV OFS-4(SP),R2 ; SWAP OLD R5 AND RETURN ADDRESS
351 000376 010566 000012 MOV R5,CFS+4(SP) ; ...
352 000402 010205 MOV R2,R5 ; ...
353 000404 RESRG <R2,R1> ; RESTORE REGISTERS
354
355 000410 CALL @(<SP>+ ; CALL THE SUBROUTINE
356
357 .IF DF K$$DAS
358 MOV (SP)+,KINAR5 ; RESTORE INSTRUCTION MAPPING
359 .ENDC ; DF K$$DAS
360
361 000412 012667 000000G MOV (SP)+,KISAR5 ; RESTORE PROCESS MAPPING
362 000416 012667 000000G MOV (SP)+,$CMPDV ; RESTORE CURRENT PDV INDEX
363 000422 RETURN

```

FILEID**CEBUF

```

CCCCCCCC EEEEEEEEE BB888888 UU UU FFFFFFFF
CCCCCCCC EEEEEEEEE BB888888 UU UU FFFFFFFF
CC EEE BB BB UU UU FF
CC EEE BB BB UU UU FF
CC EEE BB BB UU UU FF
CC EEE BB BB UU UU FF
CC EEEEE BB888888 UU UU FFFFFFFF
CC EEEEE BB888888 UU UU FFFFFFFF
CC EEE BB BB UU UU FF
CC EEE BB BB UU UU FF
CC EEE BB BB UU UU FF
CC EEE BB BB UU UU FF
CCCCCCCC EEEEEEEEE BB888888 UUUUUUUUUU FF
CCCCCCCC EEEEEEEEE BB888888 UUUUUUUUUU FF

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSSSS TT
LLLLLLLLL SSSSSSSS TT
LLLLLLLLL SSSSSSSS TT

```

410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466

000534

```
.SBTTL CCBGT - CCB allocation routine

*--CCBGT- CCB allocation routine

This subroutine is called within this module to allocate a CCB
from the CCB pool or from System Dynamic Memory. This routine is
always executed with interrupts inhibited (and multi-processors
locks set).

Inputs:
  None.

Outputs:
  R4 contains the address of allocated CCB
  C-Bit is CLEAR if the CCB was successfully allocated
  C-Bit is SET if the allocation failed

Notes:
  Before this routine is called, the calling routine must have issued
  a CLC instruction followed by an INHIB$. No other values must be
  stored on the stack because if the allocation fails, the FAIL: code
  will reach back on the stack and set the C-Bit in the saved PSW so
  that when the calling routine executes an ENABL$ the C-Bit will be
  set to indicate the allocation failure.

CCBGT: .IF DF N$$OPT

        TSTB    PRI0FF+2(SP)    ;; Can we dynamically allocate a CCB?
        BMI     10$             ;; If MI, No

        CMP     $CCBCT,$RDBTH    ;; Else, are we below the RDB threshold
        BHI     10$             ;; If HI, No ... allocate from pool

        CMP     $CCBAL,$RDBNM    ;; Have we allocated enough dynamic CCBs?
        BHS     30$             ;; If HIS, Yes

        SAVRG   <R0,R1,R2>      ;; Save some registers
        MOV     $CCBSZ,R1        ;; Get size of block to allocate

        .IF DF M$$PRO
        ADD     #2,R1            ;; 2 more bytes on a Multi-Processor
        .ENDC

        CALL    $ALOCB          ;; Try to allocate a dynamic CCB
        MOV     R0,R4           ;; Copy adress of allocated block
        RESRG   <R2,R1,R0>      ;; Restore registers
        BCS     10$             ;; If CS, DSR allocation failed

        .IF DF M$$PRO
        TST     (R4)+           ;; Skip over Unibus Run Mask
        .ENDC

        INC     $CCBAL          ;; Count dynamically allocated CCB
        BR      20$             ;; Enter common code

        .ENDC
```

```

CCCCCCCC  EEEEEEEEE 88888888 UU UU FFFFFFFF 11
CCCCCCCC  EEEEEEEEE 88888888 UU UU FFFFFFFF 11
CC         EE         88      88 UU UU FF      1111
CC         EE         88      88 UU UU FF      1111
CC         EE         88      88 UU UU FF      11
CC         EE         88      88 UU UU FF      11
CC         EEEEEEEE 88888888 UU UU FFFFFFFF 11
CC         EEEEEEEE 88888888 UU UU FFFFFFFF 11
CC         EE         88      88 UU UU FF      11
CC         EE         88      88 UU UU FF      11
CC         EE         88      88 UU UU FF      11
CC         EE         88      88 UU UU FF      11
CCCCCCCC  EEEEEEEEE 88888888 UUUUUUUUUU FF      111111
CCCCCCCC  EEEEEEEEE 88888888 UUUUUUUUUU FF      111111

```

```

LL          SSSSSSSS TTTTTTTTTT
LL          SSSSSSSS TTTTTTTTTT
LL          SS       TT
LL          SS       TT
LL          SS       TT
LL          SS       TT
LL          SSSSSS   TT
LL          SSSSSS   TT
LL          SS       TT
LL          SS       TT
LL          SS       TT
LL          SS       TT
LL          SS       TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```

```

410 .SBTTL CCBGT - CCB allocation routine
411
412 *--CCBGT- CCB allocation routine
413
414 This subroutine is called within this module to allocate a CCB
415 from the CCB pool or from System Dynamic Memory. This routine is
416 always executed with interrupts inhibited (and multi-processors
417 locks set).
418
419 Inputs:
420 None.
421
422 Outputs:
423 R4 contains the address of allocated CCB
424 C-Bit is CLEAR if the CCB was successfully allocated
425 C-Bit is SET if the allocation failed
426
427 Notes:
428 Before this routine is called, the calling routine must have issued
429 a CLC instruction followed by an INHIB$. No other values must be
430 stored on the stack because if the allocation fails, the FAIL: code
431 will reach back on the stack and set the C-Bit in the saved PSW so
432 that when the calling routine executes an ENABL$ the C-Bit will be
433 set to indicate the allocation failure.
434
435
436 CCBGT: .IF DF N$SOPT
437
438 000534 TSTB PRIOFF+2(SP)    ;;; Can we dynamically allocate a CCB?
439 000540 BMI 10$        ;;; If MI, No
440
441 000542 026767 000000G 000000G CMP $CCBCT,$RDBTH    ;;; Else, are we below the RDB threshold
442 000550 101023        ;;; If HI, No ... allocate from pool
443
444 000552 026767 000000G 000000G CMP $CCBAL,$RDBNM    ;;; Have we allocated enough dynamic CCBs?
445 000560 103033        ;;; If HI, Yes
446
447 000562 SAVRG <R0,R1,R2>    ;;; Save some registers
448 000570 016701 000000G MOV $CCBSZ,R1    ;;; Get size of block to allocate
449
450 .IF DF M$S$PRO
451 ADD #2,R1    ;;; 2 more bytes on a Multi-Processor
452 .ENDC
453
454 000574 CALL $ALOCB    ;;; Try to allocate a dynamic CCB
455 000600 010004 MOV R0,R4    ;;; Copy address of allocated block
456 000602 RESRG <R2,R1,R0>    ;;; Restore registers
457 000610 1034C3 BCS 10$    ;;; If CS, DSR allocation failed
458
459 .IF DF M$S$PRO
460 TST (R4)+    ;;; Skip over Unibus Run Mask
461 .ENDC
462
463 000612 005267 000000G INC $CCBAL    ;;; Count dynamically allocated CCB
464 000616 000407 BR 20$    ;;; Enter common code
465
466 .ENDC

```


FILEID**CEDDM

```

CCCCC      EEEEEEEEE DDDDDDDD DDDDDDDD MM      MM
CCCCC      EEEEEEEEE DDDDDDDD DDDDDDDD MM      MM
CC          EE         DD         DD DD         DD MMMM MMMM
CC          EE         DD         DD DD         DD MMMM MMMM
CC          EE         DD         DD DD         DD MM MM MM
CC          EE         DD         DD DD         DD MM MM MM
CC          EEEEEEEE DD         DD DD         DD MM MM MM
CC          EEEEEEEE DD         DD DD         DD MM MM MM
CC          EE         DD         DD DD         DD MM MM MM
CC          EE         DD         DD DD         DD MM MM MM
CC          EE         DD         DD DD         DD MM MM MM
CCCCCCCC   EEEEEEEEE DDDDDDDD DDDDDDDD MM      MM
CCCCCCCC   EEEEEEEEE DDDDDDDD DDDDDDDD MM      MM

```

```

....
....
....
....

```

```

LL          SSSSSSSS TTTTTTTTTT
LL          SSSSSSSS TTTTTTTTTT
LL          SS         TT
LL          SS         TT
LL          SS         TT
LL          SS         TT
LL          SSSSSS     TT
LL          SSSSSS     TT
LL          SS         TT
LL          SS         TT
LL          SS         TT
LL          SS         TT
LL          SS         TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```

```

424 .SBTTL $DDAST - ASYNCHRONOUS COMPLETION TO DLC LEVEL
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452 000274 010446 $DDAST:MOV R4,-(SP) ; SAVE LINE NUMBER
453 000276 $CCBGT ; ALLOCATE A CCB
454 000302 103406 BCS 5$ ; IF CS ERROR
455 000304 012664 000006 MOV (SP)+,C.LIN(R4) ; SET LINE NUMBER IN CCB
456 000310 012764 000020 000010 MOV #FC.CCP+FS.AST,C.FNC(R4) ; SET ERROR COMPLETE FUNCTION CODE
457 000316 000426 BR 20$ ; FINISH IN COMMON CODE
458
459 000320 012604 5$: MOV (SP)+,R4 ; CLEAN THE STACK
460 000322 RETURN ; RETURN TO CALLER
461

```

1 .IIF DF X\$\$NDM .TITLE CEDDMM
2 .IIF DF X\$\$MDC .TITLE CEDDMM
3 .IIF NDF X\$\$NDM & X\$\$MDC .TITLE CEDDMM
4 .IDENT /V05.00/
5
6
7
8
9

10 COPYRIGHT (C) 1978,1979,1980, 1982, 1983, 1985 BY
11 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
12
13

14 THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 TRANSFERRED.
20

21 THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 CORPORATION.
24

25 DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27

28 MODULE DESCRIPTION:
29

30 CEX DDM/DLC INTERFACE ROUTINES
31

32 DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
33

34 IDENT HISTORY:
35

- 36 1.00 10-FEB-78
37 VERSION 2.0 RELEASE
38
39 2.00 14-DEC-79
40 DECNET-11M/S V3.0
41 DECNET-11M-PLUS V1.0
42
43 3.00 16-APR-82
44 DECNET-11M V3.1
45 DECNET-11M-PLUS V1.1
46
47 4.00 07-NOV-83
48 DECNET-11M V4.0
49 DECNET-11M-PLUS V2.0
50
51 5.00 22-JUL-85
52 DECnet-11M/S V4.2
53 DECnet-11M-Plus V3.0
54 DECnet-Micro/Rsx V1.0
55
56
57

```

520      ADD    $SLTMA,R3      ; POINT INTO SYSTEM LINE INDEX TABLE
521      MOV    (R3),R3        ; GET ADDRESS OF SYSTEM LINE TABLE
522      MOV    L.KRBA(R3),R3  ; GET POINTER TO KRB
523      MOV    K.URM(R3),C.URM(R4)
524
525      .ENDC
526
527      000504      CALLR    $PDQU1      ; QUEUE CCB AND SCHEDULE PROCESS
528
529      .DSABL    LSB
530
531      .ENDC
  
```

```

71      ; LOCAL MACRO DEFINITIONS
72      ;
73      ;
74      ;
75      .IF DF K$$DAS
76      .MACRO DDFDF,FNC,ARG1
77      MOV R3,-(SP)      ; SAVE R3
78      MOV #FNC,R3      ; GET FUNCTION CODE
79      .IF B <ARG1>
80      BR DDCM1          ; COMMON PROCESSING FOR NON-CCB CONTROL FUNCTION
81      .IFF
82      .IF IDN <ARG1>,<MDC>
83      BR DDCM2          ; COMMON PROCESSING FOR MODEM CONTROL FUNCTIONS
84      .IFF
85      .IF IDN <ARG1>,<SUB>
86      BR DDCM4          ; COMMON PROCESSING FOR CONTROL FUNCTIONS WITH CCB
87      .IFF
88      BR DDCM3          ; COMMON PROCESSING FOR CONTROL FUNCTIONS WITH CCB
89      .ENDC
90      .ENDC
91      .ENDC
92      .ENDM DDFDF
93      .IFF ; DF K$$DAS
94      .MACRO DDFDF,FNC,ARG1
95      .IF B <ARG1>
96      JSR R3,DDCM1      ; COMMON PROCESSING FOR NON-CCB CONTROL FUNCTION
97      .IFF
98      .IF IDN <ARG1>,<MDC>
99      JSR R3,DDCM2      ; COMMON PROCESSING FOR MODEM CONTROL FUNCTIONS
100      .IFF
101      .IF IDN <ARG1>,<SUB>
102      JSR R3,DDCM4      ; COMMON PROCESSING FOR CONTROL FUNCTIONS WITH CCB
103      .IFF
104      JSR R3,DDCM3      ; COMMON PROCESSING FOR CONTROL FUNCTIONS WITH CCB
105      .ENDC
106      .ENDC
107      .ENDC
108      .WORD FNC
109      .ENDM
110      .ENDC ; DF K$$DAS
111
112
113
114

```

AS\$CHK= 000000	CS.ENB= 000020	FC.KIL= 000004	LD\$LP = 000000	N\$B\$UF= 000001
AS\$CPS= 000000	CS.ERR= 100000	FC.MAN= 000024	LF.ACT= 100000	N\$LDV= 000001
AS\$PRI= 000000	CS.FTL= 001000	FC.MLD= 000026	LF.BRO= 000400	N\$MCP= 000001
AS\$TRP= 000000	CS.HCR= 000001	FC.PCT= 000030	LF.BWT= 000007	N\$MLL= 000001
CB.CCB= 000002	CS.HFE= 002000	FC.PWR= 000022	LF.ENA= 002000	N\$MOV= 0000.0
CB.DDM= 000040	CS.LST= 040000	FC.RCE= 000002	LF.LPB= 001000	N\$SNC= 000001
CB.DLC= 000020	CS.MTL= 000000	FC.RCP= 000014	LF.MDC= 000100	N\$SPEN= 000001
CB.RDB= 000004	CS.RNG= 000010	FC.TIM= 000010	LF.MFL= 004000	P\$P45= 000000
CB.SDB= 000010	CS.ROV= 000004	FC.XCP= 000012	LF.MTP= 000020	P\$SWRD= 000000
CB.SLI= 000100	CS.RSN= 010000	FC.XME= 000020	LF.PAC= 000200	Q\$SOPT= 000010
CB.XLB= 000001	CS.SHU= 000001	FS.AST= 000000	LF.RDY= 040000	R\$SDER= 000000
CC.LLC= 000200	CS.SID= 000002	FS.CIB= 002000	LF.REA= 010000	R\$SK11= 000001
CE.ABO= 100362	CS.STR= 000004	FS.CRA= 001000	LF.SER= 000040	R\$SSND= 000000
CE.DAO= 100346	CS.SUC= 000001	FS.DIS= 013000	LF.TIM= 000010	R\$S11M= 000000
CE.DIS= 100366	CS.TMO= 020000	FS.DVC= 001000	LF.UNL= 020000	SF.ACT= 000200
CE.ERR= 100370	CS.XUR= 000004	FS.ENB= 012000	LF.X2P= 000000	SF.ENA= 000100
CE.ILN= 100350	C\$CKP= 000000	FS.EXI= 001000	LN.CLO= 000000	SF.LPB= 000004
CE.LTC= 100356	C\$ORE= 000400	FS.GET= 006000	LN.DUM= 000005	SF.MFL= 000040
CE.MOP= 100372	C\$RSR= 177564	FS.HLT= 000000	LN.LOA= 000004	SF.PAC= 000020
CE.PFE= 100361	C.ADD= 000034	FS.INI= 000000	LN.LOU= 000003	SF.REA= 000010
CE.RIE= 100376	C.BID= 000003	FS.KIL= 000000	LN.OAU= 000003	SF.SER= 000001
CE.SRC= 100364	C.BUF= 000014	FS.LCL= 100000	LN.OFF= 000001	SF.SVC= 000002
CE.STP= 100352	C.BUF1= 000014	FS.LTM= 001000	LN.ON= 000000	SF.UNL= 000040
CE.TME= 100354	C.BUF2= 000024	FS.MNT= 004000	LN.OOP= 000004	S\$WRG= 000000
CE.TMO= 100374	C.CNT= 000020	FS.MSN= 014000	LN.OPE= 000001	S\$YSZ= 007600
CE.UNS= 100344	C.CNT1= 000020	FS.REA= 001000	LN.REF= 000002	S.COST= 000001
CF.CHN= 000001	C.CNT2= 000030	FC.RET= 000000	LN.SER= 000002	S.FLG= 000000
CF.EOM= 000004	C.FLG= 000022	FC.REZ= 005000	LN.STA= 000017	S.LEN= 000004
CF.HDR= 000020	C.FLG1= 000022	FS.RLB= 002000	LN.SUB= 000360	S.NMST= 000002
CF.LB= 100000	C.FLG2= 000032	FS.RNG= 011000	LN.TRI= 000006	S.OWNR= 000003
CF.LIN= 000002	C.FNC= 000010	FS.RST= 000000	L\$ASG= 000000	T\$KMG= 000000
CF.SOM= 000010	C.LIN= 000006	FS.RTN= 001000	L\$PRV= 000000	T\$MIN= 000000
CF.SYN= 000040	C.LNK= 000000	FS.SET= 005000	L\$PT1= 000001	V\$CTR= 001000
CF.TRN= 000100	C.MOD= 000011	FS.SFC= 005000	L\$11R= 000000	X\$DBT= 000000
CM.CIR= 000002	C.NSP= 000004	FS.SFR= 006000	L.COST= 000015	X\$NDM= 000001
CM.FMI= 100000	C.PRO= 000042	FS.SFS= 004000	L.CTL= 000012	ZF.COU= 001000
CM.HRD= 000002	C.RSV= 000002	FS.SPW= 040000	L.CVA= 177776	ZF.DDM= 000001
CM.LIN= 000000	C.STA= 000007	FS.STM= 000000	L.DDM= 000002	ZF.DIA= 004000
CM.LOC= 000001	C.STS= 000012	FS.STP= 002000	L.DDS= 000004	ZF.DLC= 000002
CM.XLO= 000004	C.URM= 177776	FS.STR= 001000	L.DLC= 000003	ZF.DVP= 100000
CP.DCF= 000040	C.XACP= 000004	FS.TRM= 003000	L.DLM= 000006	ZF.ING= 040000
CP.HDL= 000007	C.XID= 000035	FS.WLB= 001000	L.DLS= 000010	ZF.KMX= 000020
CP.PS= 177400	C.XLEN= 000044	FS.XKL= 002000	L.FIG= 000000	ZF.LLC= 000004
CP.PSI= 000200	C.XPLI= 000040	FS.XOF= 001000	L.KRBA= 000016	ZF.LMC= 000100
CP.XCF= 000100	C.XPT= 000034	FS.XON= 007000	L.LEN= 000022	ZF.MAN= 020000
CP.ZFR= 000030	C.XSVC= 000042	FS.ZER= 002000	L.MPF= 000022	ZF.MFL= 000010
CS.ABO= 000100	C.XTC= 000037	F\$LVLL= 000001	L.NMST= 000020	ZF.MI= 000400
CS.BRO= 000002	C.X25= 000036	G\$TTP= 000000	L.NSTA= 000014	ZF.MUX= 000040
CS.BUF= 000200	D\$BUG= 177514	G\$TSS= 000000	L.OWNR= 000021	ZF.PSE= 002000
CS.CES= 000002	D\$LSK= 000000	G\$TTR= 000000	L.UNIT= 000013	ZF.SLI= 010000
CS.CHN= 000010	D\$L11= 000001	G\$WRD= 000000	M\$CRB= 000124	ZF.TIM= 000200
CS.CMP= 000200	D\$YNC= 000000	I\$RAR= 000000	M\$CRX= 000000	ZF.X3P= 000000
CS.DCR= 000400	D\$YNM= 000000	T\$PR= 000000	M\$FCS= 000000	ZS.ASN= 100000
CS.DEF= 000004	E\$XPR= 000000	K\$CNT= 177546	M\$MGE= 000000	ZS.BSV= 140000
CS.DEV= 000002	FC.CCP= 000020	K\$CSR= 177546	M\$NET= 000000	Z.AVL= 000014
CS.DIS= 000040	FC.CTL= 000006	K\$LDL= 000000	M\$OVR= 000000	Z.DAT= 000016
CS.ENA= 000001	FC.KCP= 000016	K\$TPS= 000074	N\$ACC= 000001	Z.DSP= 000000

CEDLC MACRO V05.03b Friday 28-Jun-85 18:19 Page 7-2
Symbol table

A\$\$CHK= 000000	CS.ENB= 000020	FC.KIL= 000004	LD\$LP = 000000	N\$\$BUF= 000001
A\$\$CPS= 000000	CS.ERR= 100000	FC.MAN= 000024	LF.ACT= 100000	N\$\$LDV= 000001
A\$\$PRI= 000000	CS.FTL= 001000	FC.MLD= 000026	LF.BRO= 000400	N\$\$MCP= 000001
A\$\$TRP= 000000	CS.HCR= 000001	FC.PCT= 000030	LF.BWT= 000007	N\$\$MLL= 000001
CB.CCB= 000002	CS.HFE= 002000	FC.PWR= 000022	LF.ENA= 002000	N\$\$MOV= 000010
CB.DDM= 000040	CS.LST= 040000	FC.RCE= 000002	LF.LPB= 001000	N\$\$NCT= 000001
CB.MLC= 000020	CS.MTL= 004000	FC.RCP= 000014	LF.MDC= 000100	N\$\$PEM= 000001
CB.RDB= 000004	CS.RNG= 000010	FC.TIM= 000010	LF.MFL= 004000	P\$\$P45= 000000
CB.SDB= 000010	CS.ROV= 000004	FC.XCP= 000012	LF.MTP= 000020	P\$\$WRD= 000000
CB.SLI= 000100	CS.RSN= 010000	FC.XME= 000000	LF.PAC= 000200	Q\$\$OFT= 000010
CB.XLB= 000001	CS.SHU= 000001	FS.AST= 000000	LF.RDY= 040000	R\$\$DER= 000000
CC.LLC= 000200	CS.SID= 000002	FS.CIB= 002000	LF.REA= 010000	R\$\$K11= 000001
CE.ABO= 100362	CS.STR= 000004	FS.CRA= 001000	LF.SER= 000040	R\$\$SND= 000000
CE.DAO= 100346	CS.SUC= 000001	FS.DIS= 013000	LF.TIM= 000010	R\$\$17M= 000000
CE.DIS= 100366	CS.TMO= 020000	FS.DVC= 001000	LF.UNL= 020000	SF.ACT= 000200
CE.ERR= 100370	CS.XUR= 000004	FS.ENB= 012000	LF.X2P= 000000	SF.ENA= 000100
CE.ILN= 100350	C\$\$CKP= 000000	FS.EXI= 001000	LN.CLO= 000000	SF.LPB= 000004
CE.LTO= 100356	C\$\$ORE= 000400	FS.GET= 006000	LN.DUM= 000005	SF.MFL= 000040
CE.MOP= 100372	C\$\$RSH= 177564	FS.HLT= 000000	LN.LOA= 000004	SF.PAC= 000020
CE.NTE= 100361	C.ADD= 000034	FS.INJ= 000000	LN.LOO= 000003	SF.REA= 000010
CE.RTE= 100376	C.BID= 000003	FS.KIL= 000000	LN.OAU= 000003	SF.SER= 000001
CE.SRC= 100364	C.BUF= 000014	FS.LCL= 100000	LN.OFF= 000001	SF.SVC= 000002
CE.STP= 100352	C.BUF1= 000014	FS.LTM= 001000	LN.ON= 000000	SF.UNL= 000040
CE.TMC= 100354	C.BUF2= 000024	FS.MNT= 004000	LN.OOP= 000004	S\$\$WRG= 000000
CE.TMO= 100374	C.CNT= 000020	FS.MSN= 014000	LN.OPE= 000001	S\$\$YSZ= 007600
CE.UNS= 100344	C.CNT1= 000020	FS.REA= 001000	LN.REF= 000002	S.COST= 000001
CF.CHN= 000001	C.CNT2= 000030	FS.RET= 000000	LN.SER= 000002	S.FLG= 000000
CF.EOM= 000004	C.FLG= 000022	FS.RFZ= 003000	LN.STA= 000017	S.LEN= 000004
CF.HDR= 000020	C.FLG1= 000022	FS.RLB= 002000	LN.SUB= 000360	S.NMST= 000002
CF.LB= 100000	C.FLG2= 000032	FS.RNG= 011000	LN.TRI= 000006	S.OWNR= 000003
CF.LIN= 000002	C.FNC= 000010	FS.RST= 000000	L\$\$ASG= 000000	T\$\$KMG= 000000
CF.SOM= 000010	C.LIN= 000006	FS.RTN= 001000	L\$\$DRV= 000000	T\$\$MIN= 000000
CF.SYN= 000040	C.LNK= 000000	FS.SET= 005000	L\$\$P11= 000001	V\$\$CTR= 001000
CF.TRN= 000100	C.MOD= 000011	FS.SFL= 005000	L\$\$11R= 000000	X\$\$DBT= 000000
CM.CIR= 000002	C.MSP= 000004	FS.SFR= 006000	L.COST= 000015	ZF.COU= 001000
CM.CMT= 100000	C.PRO= 000042	FS.SFS= 004000	L.CTL= 000012	ZF.DDM= 000001
CM.HRD= 000002	C.RSV= 000002	FS.SPW= 040000	L.CVA= 177776	ZF.DIA= 004000
CM.LIN= 000000	C.STA= 000007	FS.STM= 000000	L.DDM= 000002	ZF.DLC= 000002
CM.LOO= 000001	C.STS= 000012	FS.STP= 002000	L.DDS= 000004	ZF.DVP= 100000
CM.XLO= 000004	C.URM= 177776	FS.STR= 001000	L.DLC= 000003	ZF.INI= 040000
CP.DCF= 000040	C.XACP= 000004	FS.TRM= 003000	L.DLM= 000006	ZF.KMX= 000020
CP.HDL= 000007	C.XID= 000035	FS.WLB= 001000	L.DLS= 000010	ZF.LLC= 000004
CP.PS= 177400	C.XLCN= 000044	FS.XKL= 002000	L.FLG= 000000	ZF.LMC= 000100
CP.PST= 000200	C.XPLI= 000040	FS.XGF= 010000	L.KRBA= 000016	ZF.MAN= 020000
CP.XCF= 000100	C.XPT= 000034	FS.XON= 007000	L.LEN= 000022	ZF.MFL= 000010
CP.2FR= 000030	C.XSVC= 000042	FS.ZEH= 002000	L.MPF= 000022	ZF.MTM= 000400
CS.ABO= 000100	C.XTC= 000037	F\$\$LVL= 000001	L.NMST= 000020	ZF.MUX= 000040
CS.BRO= 000002	C.X25= 000036	G\$\$TPP= 000000	L.NSTA= 000014	ZF.PSE= 002000
CS.BUF= 000200	D\$\$BUG= 177514	G\$\$TSS= 000000	L.OWNR= 000021	ZF.SLI= 010000
CS.CES= 000002	D\$\$ISK= 000000	G\$\$TTK= 000000	L.UNT= 000013	ZF.TIM= 000200
CS.CHN= 000010	D\$\$L11= 000001	G\$\$WRD= 000000	M\$\$CRB= 000124	ZF.X3P= 000000
CS.CMP= 000200	D\$\$YNC= 000000	I\$\$RAR= 000000	M\$\$CRX= 000000	ZS.ASN= 100000
CS.DCR= 000400	D\$\$YNM= 000000	J\$\$RCN= 000000	M\$\$FCS= 000000	ZS.BSY= 140000
CS.DEF= 000004	E\$\$XPR= 000000	K\$\$CNT= 177546	M\$\$MGE= 000000	Z.AVL= 000014
CS.DEV= 000002	FC.CCP= 000020	K\$\$CSR= 177546	M\$\$NET= 000000	Z.DAT= 000016
CS.DIS= 000040	FC.CTL= 000006	K\$\$LDC= 000000	M\$\$OVR= 000000	Z.DSP= 000000
CS.EFA= 000001	FC.KCP= 000016	K\$\$TPS= 000074	N\$\$ACC= 000001	Z.FLG= 000010

```

182 .SBTTL PROCESS TO PROCESS DIRECT CALL INTERFACE
183
184 *--$LLCLC-PROCESS TO PROCESS DIRECT CALL INTERFACE (WITH CCB)
185 *--$LLCAL-PROCESS TO PROCESS DIRECT CALL INTERFACE
186
187 THIS ROUTINE IS CALLED BY A PROCESS (TYPICALLY AN LLC) TO DISPATCH TO
188 ANOTHER PROCESS (TYPICALLY ANOTHER LLC) VIA IT'S DISPATCH TABLE.
189
190 INPUTS:
191 R0 = OPTIONAL CALLING PARAMETER TO DESTINATION PROCESS
192 R1 =
193 R2 = PROCESS PDV INDEX ($LLCAL ONLY)
194 R3 = ADDRESS OF FUNCTION CODE ($LLCAL ONLY)
195 R4 = ADDRESS OF CCB ($LLCLC ONLY)
196 C.FNC - FUNCTION CODE
197 C.MOD - FUNCTION MODIFIER
198 C.STA - DESTINATION PDV INDEX
199
200 OUTPUTS: (TO THE DESTINATION PROCESS)
201 R3 = SUBFUNCTION CODE
202 R4 = ADDRESS OF CCB (OPTIONAL)
203 C.STA - SOURCE PDV INDEX
204 R5 = ADDRESS OF PROCESS DATABASE DESCRIPTOR
205
206 REGISTERS MODIFIED:
207 R0, R1, R2, R3, R4
208
209 NOTE:
210 If the $LLCAL call is being performed from a task in an I and D
211 space system, the word pointed to by R3 will not be accessible by $PDDSP
212 because it will be mapped via Kernel instruction APR5. To work around
213 this mapping problem, the word pointed to by R3 should be in DSR or on
214 the Kernel Stack.
215
216 $LLCLC::MOV R4,R3 ; COMPUTE ADDRESS OF FUNCTION CODE
217 ADD #C.FNC,R3 ;
218 MOVB C.STA(R4),R2 ; GET DESTINATION PDV INDEX
219 MOVB $CMPDV,C.STA(R4); SAVE SOURCE PDV INDEX
220
221 $LLCAL::SAVRG <R5> ; SAVE CURRENT DATABASE DESCRIPTOR
222 MOV R2,R5 ; COPY PDV INDEX
223 ADD $PDVTA,R5 ; POINT INTO PDV INDEX TABLE
224 MOV (R5),R5 ; GET PDV ADDRESS
225 MOV Z.DAT(R5),R5 ; GET ADDRESS OF DATABASE DESCRIPTOR
226
227 CALL $PDDSP ; DISPATCH TO THE PROCESS
228
229 RESRG <R5> ; RECOVER DATABASE DESCRIPTOR
230 RETURN
231
232 000001 .END

```


CELOG CREATED BY MACRO ON 28-JUN-85 AT 18:19 PAGE 1 C 13

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
I\$\$AS	= *****	4-54
N\$\$EVL	= *****	4-1 4-2 5-104
R\$\$11D	= *****	4-54
R\$\$11M	= 000000	4-54
R\$\$11S	= *****	4-54
X\$\$MCB	= *****	4-54 4-54
ZF.COU	= 001000	#4-54
ZF.DDM	= 000001	#4-54
ZF.DIA	= 004000	#4-54
ZF.DLC	= 000002	#4-54
ZF.DVP	= 100000	#4-54
ZF.INI	= 040000	#4-54
ZF.KMX	= 000020	#4-54
ZF.LLC	= 000004	#4-54
ZF.LMC	= 000100	#4-54
ZF.MAN	= 020000	#4-54
ZF.MFL	= 000010	#4-54
ZF.MTM	= 000400	#4-54
ZF.MUX	= 000040	#4-54
ZF.PSE	= 002000	#4-54
ZF.SLI	= 010000	#4-54
ZF.TIM	= 000200	#4-54
ZF.X3P	= 000000	#4-54
ZS.ASN	= 100000	#4-54
ZS.BSY	= 140000	#4-54
Z.AVL	000014	#4-54
Z.DAT	000016	#4-54
Z.DSP	000000	#4-54 4-54
Z.FLG	000010	#4-54
Z.LEN	= 000016	#4-54
Z.LLN	000006	#4-54
Z.MAP	000020	#4-54
Z.NAM	000004	#4-54
Z.PCB	000012	#4-54
Z.SCH	000007	#4-54
\$CELOG	000000 RG	#5-104

```
58  
59  
60  
61  
62  
63 000000  
64 000000  
65 000000  
66  
67  
68  
69  
70  
71  
72
```

```
      ; MACRO LIBRARY CALLS  
      ;  
      .MCALL INHIB$,ENABL$,SAVRG,RESRG  
      .MCALL CCBDF$,PDVDF$,OPTDF$  
      CCBDF$ ; DEFINE THE CCB OFFSETS  
      PDVDF$ ; DEFINE THE PDV OFFSETS  
      OPTDF$ <=>,<:> ; DEFINE COMM/EXEC OPTIONS  
  
      .IF DF R$$MPL  
  
      .MCALL SCBDF$ ; DEFINE SCB OFFSETS  
      SCBDF$  
  
      .ENDC
```

CESCH CREATED BY MACRO ON 28-JUN-85 AT 18:20

PAGE 2 C 15

SYMBOL CROSS REFERENCE

CREF 04.00

SYMBOL	VALUE	REFERENCES
\$CEPWR	000304 RG	#10-361
\$CMFRK	= ***** GX	9-316
\$CMPDV	= ***** GX	8-242 *8-243 *8-271
\$CXOPT	= ***** GX	9-327
\$DDFNC	= ***** GX	10-366
\$FRKHD	= ***** GX	9-339 *9-340
\$INTSX	000000 RG	#5-91
\$INTX1	= ***** GX	5-148
\$INTX7	000042 RG	#6-169
\$PDDSP	000076 RG	#8-241
\$PDQUE	000170 RG	#9-306
\$PDQU1	000174 RG	#9-308
\$PDSPL	000052 RG	#7-203 10-368
\$PDVTA	= ***** GX	7-205 8-244
\$STKDP	= ***** GX	*5-91

CESUB MACRO V05.03b Friday 28-Jun-85 18:20 Page 11
 \$CEACC - ACCESS BLOCK IN EXTENDED POOL

```

365 .SBTTL $CEACC - ACCESS BLOCK IN EXTENDED POOL
366 .SBTTL $CECAC - ACCESS BLOCK IN ALTERNATE EXTENDED POOL
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390 000424 016767 000000G 000000G $CEACC::MOV $XBIAS,KISAR6 ; MAP TO BASE ADDRESS OF POOL
391 000432 032766 000001 000002 $CECAC::BIT #1,2(SP) ; IS THE BLOCK IN EXTENDED POOL ?
392 000440 001424 BEQ 100$ ; IF EQ, NO - ALREADY MAPPED TO IT
393 000442 SAVRG <R0> ; ELSE, SAVE R0
394 000444 016600 000004 MGVR 4(SP),R0 ; COPY BLOCK ADDRESS
395 000450 042766 177701 000004 BIC #177701,4(SP) ; RETURN VIRTUAL ADDRESS
396 000456 052766 140000 000004 BIS #140000,4(SP) ; ... MAPPED THROUGH KISAR6
397
398
399
400
401
402
403
404 000006
405
406
407
408
409
410 000500 042700 176000 BIC #176000,R0 ; CLEAR EXTRANEIOUS BITS
411 000504 060067 000000G ADD R0,KISAR6 ; ... UPDATE TO PROPER BLOCK
412 000510 RESRG <R0> ; RESTORE R0
413 000512 100$: RETURN ; RETURN TO THE CALLER

```

+
 **-\$CEACC-ACCESS BLOCK IN EXTENDED POOL
 **-\$CECAC-ACCESS BLOCK IN ALTERNATE EXTENDED POOL
 THIS SUBROUTINE MAPS TO A BLOCK IN EXTENDED SINGLE WORD POOL,
 AND RETURNS A VIRTUAL ADDRESS TO ACCESS THE BLOCK. IF THE
 \$CECAC ENTRY IS CALLED, THE EXTENDED POOL MUST ALREADY BE MAPPED
 VIA APR6.
 INPUTS:
 2(SP) = ADDRESS OF BLOCK TO BE ACCESSED
 KISAR6 MAPPED TO ALTERNATE EXTENDED POOL BASE (\$CECAC)
 CUTPUTS:
 2(SP) = VIRTUAL ADDRESS OF MAPPED BLOCK
 NOTE:
 KISAR6 ALWAYS MAPPED INTO EXTENDED POOL ON EXIT EVEN IF BLOCK
 IS IN DSR.
 -

5-	56	Define Stack Offsets and Common Buffer Return Table
6-	76	\$CCBGT - Allocate a Standard CCB
7-	109	\$LDBGT - Get a Large Data Buffer
8-	137	\$RDBGT - Get a Receive Data Buffer
9-	188	\$RDBRT - Return a Receive Data Buffer
9-	189	\$LDBRT - Return a Large Data Buffer
10-	230	\$CCBRT - Return a CCB
11-	258	\$CSBGT - Get a CCB and a Small Data Buffer
12-	300	\$CSBRT - Return a CCB and a Small Data Buffer
13-	342	\$RDBWT - Queue a request for a Receive Data Buffer
14-	376	\$RDBQP - Purge Buffer Wait Queue
15-	410	CCBGT - CCB allocation routine
16-	485	CCBRT - CCB Deallocation Routine
17-	537	Allocation Control Block Documentation
18-	548	BUFGT - General Buffer Allocation Routine
19-	600	BUFRT - General Buffer Deallocation Routine

```

467
468 000534 016704 000000G      10$:  MOV    $CCBLH,R4      ::: Get address of first CCB in list
469 000540 001411              BEQ    30$              ::: If EQ, list is empty
470 000542 011467 000000G      MOV    (R4),$CCBLH      ::: Unlink CCB from list
471 000546 005367 000000G      DEC    $CCBCT          ::: Reduce count of available CCBs
472
473 000552 112764 000002 000003 20$:  MOVB   #CB.CCB,C.BID(R4) ::: Set up buffer ID byte
474
475                                .IF DF M$SPRO
476                                CLR    C.URM(R4)      ::: Allow process to run on any processor
477                                .ENDC
478
479 000560 005014              CLR    (R4)              ::: Zero next pointer (clear C-Bit)
480 000562              RETURN          ::: Return to caller
481
482 000564 005267 000000G      30$:  INC    $CCBAF          ::: Count allocation failures
483 0C0570 000433              BR      FAIL              ::: Enter common failure processing code

```

6-	56	Define Stack Offsets and Common Buffer Return Table
7-	76	\$CCBGT - Allocate a Standard CCB
8-	109	\$LDBGI - Get a Large Data Buffer
9-	137	\$RDBGI - Get a Receive Data Buffer
10-	188	\$RDBRT - Return a Receive Data Buffer
10-	189	\$LDBRT - Return a Large Data Buffer
11-	230	\$CCBRT - Return a CCB
12-	258	\$CSBGT - Get a CCB and a Small Data Buffer
13-	300	\$CSBRT - Return a CCB and a Small Data Buffer
14-	342	\$RDBWT - Queue a request for a Receive Data Buffer
15-	376	\$RDBQP - Purge Buffer Wait Queue
16-	410	CCBGT - CCB allocation routine
17-	485	CCBRT - CCB Deallocation Routine
18-	537	Allocation Control Block Documentation
19-	548	BUFGT - General Buffer Allocation Routine
20-	600	BUFRT - General Buffer Deallocation Routine

```

467
468 000620 016704 000000G      10$: MOV    $CCPLH,R4      ;;: Get address of first CCB in list
469 000624 001411              BEQ    30$              ;;: If EQ, list is empty
470 000626 011467 000000G      MOV    (R4),$CCBLH      ;;: Unlink CCB from list
471 000632 005367 000000G      DEC    $CCBCI          ;;: Reduce count of available CCBs
472
473 000636 112764 000002 000003 20$: MOVB   #CB.CCB,C.BID(R4) ;;: Set up buffer ID byte
474
475                      .IF DF M$$PRO
476                      CLR    C.URM(R4)      ;;: Allow process to run on any processor
477                      .ENDC
478
479 000644 005014              CLF    (R4)          ;;: Zero next pointer (clear C-Bit)
480 000646              RETURN                ;;: Return to caller
481
482 000650 005267 000000G      30$: INC    $CCBAF          ;;: Count allocation failures
483 000654 000470              BR      FAIL        ;;: Enter common failure processing code

```


CEDDM MACRO V05.03b Friday 28-Jun-85 18:17
 Table of contents

6-	117	\$DD??? - DLC TO DDM REQUESTS
8-	196	DDCM1 - COMMON PROCESS FOR SPECIAL NON-CCB FUNCTIONS
9-	236	DDCM2 - COMMON PROCESS FOR MODEM CONTROL FUNCTIONS
10-	317	DDCM3 - COMMON PROCESS FOR FUNCTIONS WITH CCB
10-	318	DDCM4 - COMMON PROCESS FOR FUNCTIONS WITH CCB
11-	376	DDMDSP - DISPATCH TO DDM LEVEL
12-	424	\$DDAST - ASYNCHRONOUS COMPLETION TO DLC LEVEL
13-	463	\$DDXMP - TRANSMIT COMPLETE TO DLC LEVEL
13-	464	\$DDRCP - RECEIVE COMPLETE TO DLC LEVEL
13-	465	\$DDCCP - CONTROL COMPLETE TO DLC LEVEL
13-	466	\$DDKCP - KILL COMPLETE TO DLC LEVEL
14-	533	\$STDDM - SET DDM PDV INDEX AND LINE TABLE ADDRESS

```

463 .SBTTL $DDXMP - TRANSMIT COMPLETE TO DLC LEVEL
464 .SBTTL $DDRCF - RECEIVE COMPLETE TO DLC LEVEL
465 .SBTTL $DDCCP - CONTROL COMPLETE TO DLC LEVEL
466 .SBTTL $DDKCP - KILL COMPLETE TO DLC LEVEL
467
468 ***$DDXMP-TRANSMIT COMPLETE TO DATA LINK CONTROL
469 ***$DDRCF-RECEIVE COMPLETE
470 ***$DDCCP-CONTROL COMPLETE
471 ***$DDKCP-KILL COMPLETE
472 ***$LLCSP-SPECIAL ENTRY POINT FOR X25
473
474 THIS SUBROUTINE IS CALLED BY DEVICE DRIVERS TO QUEUE
475 COMPLETION NOTIFICATIONS TO DATA LINK CONTROL MODULES.
476
477 THE $LLCSP ENTRY POINT IS PROVIDED FOR USE BY THE X25 DATA LINK MAPPING.
478
479 INPUTS:
480
481 R3 = OPERATION COMPLETION STATUS
482 R4 = ADDRESS OF CCB TO QUEUE
483 THE CCB CONTAINS A VALID LINE NUMBER
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
  $DDXMP::MOVB #FC.XCP,C.FNC(R4) ; SET TRANSMIT COMPLETE FUNCTION CODE
  BR 20$ ; JOIN COMMON CODE
  $DDRCF::MOVB #FC.RCP,C.FNC(R4) ; SET RECEIVE COMPLETE FUNCTION CODE
  BR 20$ ; JOIN COMMON CODE
  $DDKCP::MOVB #FC.KCP,C.FNC(R4) ; SET KILL COMPLETE FUNCTION CODE
  BR 20$ ; JOIN COMMON CODE
  $DDCCP::TST C.LIN(R4) ; SPECIAL MODEM CONTROL COMPLETION?
  BPL 10$ ; NO
  CALLR $CCBR* ; YES - RETURN CCB TO POOL
  ; MODEM CONTROLLER WILL POST COMPLETION
  10$: MOVB #FC.C VC(R4) ; SET CONTROL COMPLETE FUNCTION CODE
  $LLCSP::
  $DDSPC::
  20$: MOV R3,C.SIS(R4) ; C. COMPLETION STATUS
  .IF C# M$*PRO
  MOVB C.LIN(R4),R3 ; GET SYSTEM LINE #
  ASL R3 ; FORM WORD OFFSET

```

```
58
59
60
61      ; MACRO LIBRARY CALLS
62      ;
63      .MCALL INHIB$,ENABL$,SAVRG,RESRG
64      .MCALL CCBDF$,PDVDF$,SLTDF$
65      .MCALL CALLR      ; AVOID SYSTEM DEPENDENCY
66      CCBDF$           ; DEFINE THE CCB OFFSETS
67      PDVDF$           ; DEFINE THE PDV OFFSETS
68      SLTDF$           ; DEFINE THE SLT OFFSETS
69
```

```

533 .SBTTL $STDDM - SET DDM PDV INDEX AND LINE TABLE ADDRESS
534
535
536 ***$STDDM-SET DDM PDV INDEX AND LINE TABLE ADDRESS
537 ***$STDD1-(ALTERNATE ENTRY)
538
539 THIS SUBROUTINE IS CALLED TO SET UP A DEVICE DRIVER PDV INDEX AND
540 LINE TABLE ADDRESS BASED ON A SYSTEM LINE NUMBER.
541
542 INPUTS:
543
544 R2 = SYSTEM LINE NUMBER (ALTERNATE ENTRY ONLY)
545 R4 = CCB ADDRESS WITH A VALID SYSTEM LINE NUMBER IN
546 C.LIN (MAIN ENTRY ONLY)
547
548 OUTPUTS:
549
550 R2 = PDV INDEX
551 R5 = ADDRESS OF DEVICE LINE TABLE
552
553 REGISTERS MODIFIED:
554
555 R5
556
557
558 000510 $STDDM::IF NDF N$$1LN
559
560 000510 116402 000006 MOV B C.LIN(R4),R2 ; EXTRACT SYSTEM LINE NUMBER
561
562 .ENDC
563
564 000514 $STDD1::IF DF N$$1LN
565
566 MOV @$$SLTMA,R2 ; GET ADDRESS OF SYSTEM LINE ENTRY
567
568 .IFF
569
570 000514 006302 ASL R2 ; FORM WORD INDEX
571 000516 066702 ADD $$SLTMA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE
572 000522 011202 MOV (R2),R2 ; GET ADDRESS OF SYSTEM LINE ENTRY
573
574 .ENDC
575
576 000524 016205 MOV L.DDS(R2),R5 ; GET DEVICE LINE TABLE ADDRESS
577 000530 116202 MOV B L.DDM(R2),R2 ; GET DEVICE DRIVER PDV INDEX (WORD INDEX)
578 000534 RETURN ; RETURN
579
580
581 000001 .END

```

```

116      .IF NDF X$$$NDM
117      .SBTTL $$$D??? - DLC TO DDM REQUESTS
118
119      +
120      GENERAL REQUESTS WITH CCB
121      :
122      FORMAT OF CALL:
123      CALL $$$D???
124
125      INPUTS:
126      R4 = ADDRESS OF FIRST CCB IN CHAIN
127      (ALL CCBs MUST CONTAIN A VALID SLN)
128
129      OUTPUTS TO DDM:
130      R4 = ADDRESS OF FIRST CCB IN CHAIN
131      R5 = ADDRESS OF DDM LINE TABLE
132      R2 & R3 - AVAILABLE FOR USE WITHOUT SAVING
133
134      REGISTERS ACROSS CALL:
135      R0,R1 - MUST BE PRESERVED BY DDM IF USED
136      R2,R3,R5 - PRESERVED BY COMM EXEC
137      R4 - MAY BE MODIFIED
138
139      ON RETURN TO DLC:
140      C-BIT CLEAR - REQUEST HAS COMPLETED SYNCHRONOUSLY
141      R4 = ADDRESS OF CCB
142
143      C-BIT SET - REQUEST WILL COMPLETE ASYNCHRONOUSLY
144      R4 = 0
145
146      -
147
148      +
149      SPECIAL REQUESTS WITHOUT CCB
150
151      FORMAT OF CALL:
152      CALL $$$D???
153
154      INPUTS:
155      R3 = SYSTEM LINE NUMBER
156      R4 = OPTIONAL CALLING PARAMETER TO DDM
157
158      OUTPUTS TO DDM:
159      R4 = OPTIONAL CALLING PARAMETER FROM DLC
160      R5 = ADDRESS OF DDM LINE TABLE
161      R2 & R3 - AVAILABLE FOR USE WITHOUT SAVING
162
163      REGISTERS ACROSS CALL:
164      R0,R1 - MUST BE PRESERVED BY DDM IF USED
165      R2,R3,R5 - PRESERVED BY COMM EXEC
166      R4 - MAY BE MODIFIED
167
168      ON RETURN TO DLC:
169      C-BIT ALWAYS CLEAR
170      R4 = OPTIONAL RETURNING PARAMETER FROM DDM
171
172      -

```

CEDDMN MACRO V05.03b Friday 28-Jun-85 18:18 Page 15-2
Symbol table

D 10

Z.FLG	000010	Z.MAP	000020	Z.PCB	000012	\$SLTMA=	***** GX	\$STDD1	000004RG
Z.LEN	= 000016	Z.NAM	000004	Z.SCH	000007	\$STDDM	000000RG	.\$\$\$\$.	= 000034
Z.LLN	000006								

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)
000026 001 (RW,I,LCL,REL,CON)
Errors detected: 0

*** Assembler statistics

Work file reads: 8
Work file writes: 11
Size of work file: 17456 Words (69 Pages)
Size of core pool: 17608 Words (67 Pages)
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:16.18
SY: CEDDMN.V2,[130,134]CEDDMN/CR/~SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]V2,CEDDMN

CEDLC MACRO V05.03h Friday 28-Jun-85 18:19 Page 1-3
Symbol table

D 11

Z.LEN = 000016	Z.PCB 000012	\$CTCMP 000066RG	\$RCCMP 000106RG	\$STD1 000004RG
Z.LLN 000006	Z.SCH 000007	\$KLCMP 000076RG	\$SLTMA= ***** GX	\$XMCMP 000056RG
Z.MAP 000020	\$ASCMP 000034RG	\$LLCTA= ***** GX	\$STDLC 000000RG	.\$\$\$\$ = 000034
Z.NAM 000004	\$CCBGT= ***** GX	\$PDQUE= ***** GX		

. ABS. 177776 000 (RW,I,GBL,ABS,OVN)
000176 001 (RW,I,LCL,REL,CON)
Errors detected: 0

*** Assembler statistics

Work file reads: 8
Work file writes: 10
Size of work file: 17269 Words (68 Pages)
Size of core pool: 17608 Words (67 Pages)
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:12.57

SY: CEDLC.V2,[130,134]CEDLC/CP/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]CEDLC

ASSCHK= 000000	CS.ENB= 000020	FC.KIL= 000004	LD\$LP = 000000	N\$SBUF= 000001
ASSCPS= 000000	CS.ERR= 100000	FC.MAN= 000024	LF.ACT= 100000	N\$SLDV= 000001
ASSPRI= 000000	CS.FTL= 001000	FC.MLD= 000026	LF.BRO= 000400	N\$SMCP= 000001
ASSTRP= 000000	CS.HCR= 000001	FC.PCT= 000030	LF.BWT= 000007	N\$SMML= 000001
CB.CCB= 000002	CS.HFE= 002000	FC.PWR= 000022	LF.ENA= 002000	N\$SMOV= 000010
CB.DDM= 000040	CS.LST= 040000	FC.RLE= 000002	LF.LPB= 001000	N\$SNCT= 000001
CB.DLC= 000020	CS.MTL= 004000	FC.RCP= 000014	LF.MDC= 000100	N\$SPEM= 000001
CB.RDB= 000004	CS.RNG= 000010	FC.TIM= 000010	LF.MFL= 004000	PS = ***** GX
CB.SDB= 000010	CS.ROV= 000004	FC.XCP= 000012	LF.MTP= 000020	P\$SP45= 000000
CB.SLI= 000100	CS.RSN= 010000	FC.XME= 000000	LF.PAC= 000200	P\$SWRD= 000000
CB.XLB= 000001	CS.SHU= 000001	FS.AST= 000000	LF.RDY= 040000	Q\$SOP= 000010
CC.LLC= 000200	CS.SID= 000002	FS.CJB= 002000	LF.REA= 010000	R\$SDER= 000000
CE.ABO= 100362	CS.STR= 000004	FS.CRA= 001000	LF.SER= 000040	R\$SK11= 000001
CE.DAO= 100346	CS.SUC= 000001	FS.DJS= 013000	LF.TIM= 000010	R\$SSND= 000000
CE.DIS= 100366	CS.TMO= 020000	FS.DVC= 001000	LF.UNL= 020000	R\$11M= 000000
CE.ERR= 100370	CS.XUR= 000004	FS.ENB= 012000	LF.X2P= 000000	SF.ACT= 000200
CE.ILN= 100350	C\$CKP= 000000	FS.EXI= 001000	LN.CLO= 000000	SF.ENA= 000100
CE.LTO= 100356	C\$SORE= 000400	FS.GET= 006000	LN.DUM= 000005	SF.LPB= 000004
CE.MOP= 100372	C\$SRSH= 177564	FS.HLT= 000000	LN.LOA= 000004	SF.MFL= 000040
CE.NTE= 100361	C.ADD= 000034	FS.INJ= 000000	LN.LOO= 000003	SF.PAC= 000020
CE.RTE= 100376	C.BID= 000003	FS.KIL= 000000	LN.OAU= 000003	SF.REA= 000010
CE.SRC= 100364	C.BUF= 000014	FS.LCL= 100000	LN.OFF= 000001	SF.SER= 000001
CE.STP= 100352	C.BUF1= 000014	FS.LTM= 001000	LN.ON= 000000	SF.SVC= 000002
CE.TME= 100354	C.BUF2= 000024	FS.MNT= 004000	LN.OOP= 000004	SF.UNL= 000040
CE.TMO= 100374	C.CNT= 000020	FS.MSN= 014000	LN.OPE= 000001	S\$SWRG= 000000
CE.UNS= 100344	C.CNT1= 000020	FS.REA= 001000	LN.REF= 000002	S\$YST= 007600
CF.CHN= 000001	C.CNT2= 000030	FS.RET= 000000	LN.SER= 000002	S.COST= 000001
CF.EOM= 000004	C.FLG= 000022	FS.REZ= 003000	LN.STA= 000017	S.FLG= 000000
CF.HDR= 000020	C.FLG1= 000022	FS.RLB= 002000	LN.SUB= 000360	S.LEN= 000004
CF.LB= 100000	C.FLG2= 000032	FS.RNG= 011000	LN.TRI= 000006	S.NMST= 000002
CF.LIN= 000002	C.FNC= 000010	FS.RST= 000000	L\$ASG= 000000	S.OWNR= 000003
CF.SOM= 000010	C.LIN= 000006	FS.RTN= 001000	L\$DRV= 000000	T\$KMG= 000000
CF.SYN= 000040	C.LNK= 000000	FS.SET= 005000	L\$P11= 000001	T\$MIN= 000000
CF.TRN= 000100	C.MOD= 000011	FS.SFC= 005000	L\$11R= 000000	V\$CTR= 001000
CM.CIR= 000002	C.NSP= 000004	FS.SFR= 006000	L.COST= 000015	X\$DBT= 000000
CM.FMT= 100000	C.PRO= 000042	FS.SFS= 004000	L.CTL= 000012	ZF.COU= 001000
CM.HRD= 000002	C.RSV= 000002	FS.SPW= 040000	L.CVA= 177776	ZF.DDM= 000001
CM.LIN= 000000	C.STA= 000007	FS.STM= 000000	L.DDM= 000002	ZF.DIA= 004000
CM.LOO= 000001	C.STS= 000012	FS.STP= 002000	L.DDS= 000004	ZF.DLC= 000002
CM.XLO= 000004	C.URM= 177776	FS.STR= 001000	L.DLC= 000003	ZF.DVP= 100000
CP.DCF= 000040	C.YACP= 000004	FS.TRM= 003000	L.DLM= 000006	ZF.INI= 040000
CP.HDL= 000007	C.XID= 000035	FS.WLB= 001000	L.DLS= 000010	ZF.KMX= 000020
CP.PS= 177400	C.XLEN= 000044	FS.XKL= 002000	L.FLG= 000000	ZF.LLC= 000004
CP.PSI= 000200	C.XPLI= 000040	FS.XOF= 010000	L.KRBA= 000016	ZF.LMC= 000100
CP.XCF= 000100	C.XPT= 000034	FS.XON= 007000	L.LEN= 000022	ZF.MAN= 020000
CP.2FR= 000030	C.XSVC= 000042	FS.ZER= 002000	L.MPF= 000022	ZF.MFL= 000010
CS.ABO= 000100	C.XTC= 000037	F\$LVL= 000001	L.NMS= 000020	ZF.MTM= 000400
CS.BRO= 000002	C.X25= 000036	G\$TPP= 000000	L.NSTA= 000014	ZF.MUX= 000040
CS.BUF= 000200	D\$BUBG= 177514	G\$TSS= 000000	L.OWNR= 000021	ZF.PSE= 002000
CS.CES= 000002	D\$ISK= 000000	G\$TTK= 000000	L.UNT= 000013	ZF.SLI= 010000
CS.CHN= 000010	D\$111= 000001	G\$SWRD= 000000	M\$CRB= 000124	ZF.TIM= 000200
CS.CMP= 000200	D\$YNC= 000000	I\$RAR= 000000	M\$CRX= 000000	ZF.X3P= 000000
CS.DCR= 000400	D\$YNM= 000000	J\$RDN= 000000	M\$FCS= 000000	ZS.10000
CS.DEF= 000004	E\$XPR= 000000	K\$CNT= 177546	M\$MGE= 000000	ZS.BSY= 140000
CS.DEV= 000002	FC.CCP= 000020	K\$CSR= 177546	M\$NET= 000000	Z.AVL= 000014
CS.DTS= 000040	FC.CTL= 000006	K\$SLDC= 000000	M\$OVR= 000000	Z.DAT= 000016
CS.ENA= 000001	FC.KCP= 000016	K\$TPS= 000074	N\$ACC= 000001	Z.DSP= 000000

CELOG CREATED BY MACRO ON 28-JUN-85 AT 18:19 PAGE 2 D 13
MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES
PDVDF\$	#4-52 4-54
RESRG	#4-52
RETURN	5-125
SAVRG	#4-52

```

74      .SBTTL $INTSX - COMM EXEC DEVICE INTERRUPT SAVE ROUTINE
75
76      *
77      **-$INTSX-COMM EXEC DEVICE INTERRUPT SAVE ROUTINE
78
79      CALLED BY: JSR R4,$INTSX
80
81      Inputs:
82      R5 = Pointer to priority specification in device line table
83
84      Outputs:
85      The System Stack Depth is decremented, and if zero a switch
86      to the System Stack is effected.
87
88      R5 = Pointer to word following device priority
89
90      -
91      000000 005367 000000G $INTSX::DEC $STKDP      ;; Decrement Stack Depth Indicator
92
93      .IF DF L$$S11
94      TST (R5)+      ;; Point past device priority complement
95      .IFF
96
97      000004 042567 000000G BIC (R5)+,PS      ;; Lower priority to device level
98      .ENDC
99
100     .IF DF M$$PRO
101     CACHE$ SAVE      ;; Save bypass state of cache
102     .ENDC
103
104     000010 CALL @R4      ;; Call caller back
105
106     .IF DF M$$PRO
107     CACHE$ UNSAVE      ;; Restore bypass state of cache
108     .ENDC
109
110     000012 MTPS #PR7      ;; Inhibit Interrupts
111
112     .IF DF K$$DAS      ;; Slightly different for RSX-11M+
113     ;; with Kernel Data Space enabled
114     MOV 4(SP),KINAR5      ;; Restore previous I-Space APR Contents
115     MOV 6(SP),KISAR5      ;; Restore previous D-Space APR Contents
116     ;; Current contents of stack:
117     ;; R4 (Saved by JSR R4,$INTSX in DDM)
118     ;; R5 (Saved by JSR R5,'ISR' in Line Table)
119     ;; KINAR5 (Saved in Line Table)
120     ;; KISAR5 (Saved in Line Table)
121     ;; PS (Saved by Interrupt)
122     ;; PC (Saved by Interrupt)
123     MOV (SP)+,2(SP)      ;; Move saved R4 down on stack
124     MOV (SP)+,2(SP)      ;; Move saved R5 down on stack
125     ;; Contents of stack (necessary for $INTXX)
126     ;; R4 (Saved by JSR R4,$INTSX in DDM)
127     ;; R5 (Saved by JSR R5,'ISR' in Line Table)
128     ;; PS (Saved by Interrupt)
129     ;; PC (Saved by Interrupt)
130     .IFF      ;; DF K$$DAS

```

CESCH CREATED BY MACRO ON 28-JUN-85 AT 18:20

PAGE 3 D 15

MACRO CROSS REFERENCE

CREF 04.00

MACRO NAME	REFERENCES
CALL	5-104 8-262 10-368
LCBDF\$	#4-62 4-63
ENARL\$	#4-61 9-348
INH13\$	#4-61 9-317
MTP\$	5-110 7-203
OPTDF\$	4-62 4-65
PDVDF\$	#4-62 4-64
RESRG	#4-61
RETURN	8-272 9-350 10-374
SAVRG	#4-61

CESUB MACRO V05.03b Friday 28-Jun-85 18:20 Page 12
 \$PDVID - PROCESS NAME TO PDV INDEX

.SBITL \$PDVID - PROCESS NAME TO PDV INDEX

;*-- \$PDVID - PROCESS NAME TO PDV INDEX

THIS SUBROUTINE MAPS A PROCESS NAME (UP TO THREE CHARACTERS IN RAD50)
 INTO A PDV INDEX.

INPUTS:
 R2 = PROCESS NAME (RAD50)

OUTPUTS:
 R2 = PDV INDEX

```

$PDVID::SAVRG <R0,R1>      ; SAVE REGISTERS
                MOV $PDVTA,R0 ; GET ADDRESS OF PDV ADDRESS TABLE
                MOV $PDVNM,-(SP) ; GET TOTAL NUMBER OF PDVS IN SYSTEM
10$:            MOV (R0)+,R1 ; GET ADDRESS OF PDV
                BEQ 15$ ; PROCESS NOT LOADED - GO TO NEXT ONE
                CMP Z,NAM(R1),R2 ; DOES PROCESS NAME MATCH?
                BEQ 20$ ; YES - GO CALCULATE INDEX
15$:            DEC (SP) ; DECREMENT COUNT - DONE LOOPING?
                BNE 10$ ; NO
                SEC ; YES - NO MATCH
                BR 25$ ; RETURN WITH C-BIT SET

20$:            MOV R0,R2 ; GET CURRENT POSITION IN PDV ADDRESS TABLE
                SUB $PDVTA,R2 ; SUBTRACT STARTING ADDRESS OF TABLE
                TST -(R2) ; GET PROCESS PDV INDEX
25$:            INC (SP)+ ; CLEAN UP STACK AND DON'T CHANGE C-BIT
                RESRG ; RESTORE REGISTERS
                RETURN
;

```

```

415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430 000514
431 000520 016700 000000G
432 000524 016746 000000G
433 000530 012001
434 000532 001403
435 000534 026102 000000/
436 000540 001404
437 000542 005316
438 000544 001371
439 000546 000261
440 000550 000404
441
442 000552 010002
443 000554 166702 000000G
444 000560 005742
445 000562 005226
446 000564
447 000570
448

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```
.IIF NDF N$$OPT .TITLE CEBUF - CEX BUFFER MANAGEMENT ROUTINES
.IIF DF N$$OPT .TITLE CEBUF1 - CEX BUFFER MANAGEMENT ROUTINES (DYNAMIC)
.IDENT /V05.00/
```

```
..COPYRIGHT (C) 1980, 1982, 1983, 1985 BY
..DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
```

```
..THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
..SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
..INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR
..ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE
..MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH
..SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE
..TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN
..IN DEC.
```

```
..THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
..NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
..EQUIPMENT CORPORATION.
```

```
..DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
..ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
```

``` ..MODULE DESCRIPTION ```

```
..      CEX BUFFER ALLOCATION ROUTINES
```

```
..DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
```

``` ..IDENT HISTORY: ```

- 1.00 10-FEB-78
 VERSION 2.0 RELEASE
- 2.00 14-DEC-79
 DECNET-11M/S V3.0
 DECNET-11M-PLUS V1.0
- 3.00 16-APR-82
 DECNET-11M V3.1
 DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83
 DECNET-11M V4.0
 DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85
 DECnet-11M/S V4.2
 DECnet-11M-Plus V3.0
 DECnet-Micro/RSX V1.0

```

485 .SBTTL CCBRT - CCB Deallocation Routine
486
487
488 *--CCBRT- CCB Deallocation Routine
489
490 This subroutine is called within this module to deallocate a CCB
491 to the CCB pool or to System Dynamic Memory. This routine is always
492 executed with interrupts inhibited (and Multi-Processor locks set).
493
494 Inputs:
495 R4 contains the address of the CCB to release
496
497 Outputs:
498 None.
499
500
501 000572 CCBRT: .IF DF N$SOPT
502
503 .IF NDF R$SMPL
504 CMP R4,#$CEAVL ;;; Is CCB from the internal pool?
505 BHIS 10$ ;;; If HIS, Yes
506 .ENDC
507
508 ISTB PRIOFF+2(SP) ;;; Were we called from Device Priority Level?
509 BMI 10$ ;;; If MI, Yes ... can't interlock $DEACB
510
511 CMP $CCBCT,$RDBTH ;;; Can we satisfy RDB requests?
512 BLOS 10$ ;;; If LOS, No ... return to list
513 TST $CCBAL ;;; Any dynamic CCBs allocated?
514 BEQ 10$ ;;; If EQ, No ... return to list
515
516 SAVRG <R0,R1,R2,R3> ;;; Save some registers
517 MOV R4,R0 ;;; Copy address of CCB
518 MOV $CCBSZ,R1 ;;; Get size of block to deallocate
519
520 .IF DF M$SPRO
521 ADD #2,R1 ;;; 2 more bytes on Multi-Processors
522 TST -(R0) ;;; Backup over Unibus Run Mask
523 .ENDC
524
525 CALL $DEACB ;;; Deallocate the CCB
526 RESRG <R3,R2,R1,R0> ;;; Restore the registers
527 DEC $CCBAL ;;; Decrement count of dynamic CCB's
528 RETURN
529
530 .ENDC
531
532 000572 016714 000000G 10$: MOV $CCBLH,(R4) ;;; Set first pointer in returned CCB
533 000576 010467 000000G MOV R4,$CCBLH ;;; Set new first pointer
534 000602 005267 000000G INC $CCBCT ;;; Increment count of CCB's in pool
535 000606 RETURN

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```
.IIF NDF N$SOPT .TITLE CEBUF - CEX BUFFER MANAGEMENT ROUTINES
.IIF DF N$SOPT .TITLE CEBUF1 - CEX BUFFER MANAGEMENT ROUTINES (DYNAMIC)
.IDENT /V05.00/
```

```
: COPYRIGHT (C) 1980, 1982, 1983, 1985 BY
: DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
```

```
: THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
: SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
: INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR
: ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE
: MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH
: SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE
: TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN
: IN DEC.
```

```
: THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
: NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
: EQUIPMENT CORPORATION.
```

```
: DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
: ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
```

```
: MODULE DESCRIPTION
```

```
: CEX BUFFER ALLOCATION ROUTINES
```

```
: DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
```

```
: IDENT HISTORY:
```

- ```
: 1.00 10-FEB-78
: VERSION 2.0 RELEASE
:
: 2.00 14-DEC-79
: DECNET-11M/S V3.0
: DECNET-11M-PLUS V1.0
:
: 3.00 16-APR-82
: DECNET-11M V3.1
: DECNET-11M-PLUS V1.1
:
: 4.00 07-NOV-83
: DECNET-11M v4.0
: DECNET-11M-PLUS v2.0
:
: 5.00 22-JUL-85
: DECnet-11M/S V4.2
: DECnet-11M-Plus V3.0
: DECnet-Micro/R SX V1.0
```

```

485 .SBTTL CCBRT - CCB Deallocation Routine
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501 000656 CCBRT: .IF DF N$SOP7
502
503
504 000656 020427 000000G .IF NDF R$SMPL
505 000662 103032 CMP R4,#$CEAVL ;;; Is CCB from the internal pool?
506 .ENDC BHIS 10$;;; If HIS, Yes
507
508 000664 105766 000002 ISTB PRIOFF+2(SP) ;;; Were we called from Device Priority Level?
509 000670 100427 BMI 10$;;; If MI, Yes ... can't interlock $DEACB
510
511 000672 026767 000000G 000000G CMP $CCBCT,$RDBTH ;;; Can we satisfy RDB requests?
512 000700 101423 BLOS 10$;;; If LOS, No ... return to list
513 000702 005767 000000G TST $CCBAL ;;; Any dynamic CCBs allocated?
514 000706 001420 BEQ 10$;;; If EQ, No ... return to list
515
516 000710 SAVRG <R0,R1,R2,R3> ;;; Save some registers
517 000720 010400 MOV R4,R0 ;;; Copy address of CCB
518 000722 016701 000000G MOV $CCBSZ,R1 ;;; Get size of block to deallocate
519
520
521 .IF DF M$SPRO
522 ADD #2,R1 ;;; 2 more bytes on Multi-Processors
523 TST -(R0) ;;; Backup over Unibus Run Mask
524 .ENDC
525
526 000726 CALL $DEACB ;;; Deallocate the CCB
527 000732 RESRG <R3,R2,R1,R0> ;;; Restore the registers
528 000742 005367 000000G DEC $CCBAL ;;; Decrement count of dynamic CCB's
529 000746 RETURN
530
531 .ENDC
532
533 000750 016714 000000G 10$: MOV $CCBLH,(R4) ;;; Set first pointer in returned CCB
534 000754 010467 000000G MOV R4,$CCBLH ;;; Set new first pointer
535 000760 005267 000000G INC $CCBCT ;;; Increment count of CCB's in pool
536 000764 RETURN

```



1 .IIF DF X\$\$NDM .TITLE CEDDMN  
2 .IIF DF X\$\$MDC .TITLE CEDMM  
3 .IIF NDF X\$\$NDM & X\$\$MDC .TITLE CEDDM  
4 .IDENT /V05.00/  
5  
6  
7  
8  
9

10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
COPYRIGHT (C) 1978,1979,1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

CEX DDM/DLC INTERFACE ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 10-FFB-78  
VERSION 2.0 RELEASE
- 2.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

```

520 ADD $SLTMA,R3 ; POINT INTO SYSTEM LINE INDEX TABLE
521 MOV (R3),R3 ; GET ADDRESS OF SYSTEM LINE TABLE
522 MOV L.KRBA(R3),R3 ; GET POINTER TO KRB
523 MOV K.URM(R3),C.URM(R4)
524
525 .ENDC
526
527 000400 CALLR $PDQU1 ; QUEUE CCB AND SCHEDULE PROCESS
528
529 .DSABL LSB
530
531 .ENDC

```

```

71 ; LOCAL MACRO DEFINITIONS
72 ;
73 ;
74 ;
75 .IF DF K$$DAS
76 ;
77 .MACRO DD^DF,FNC,ARG1
78 MOV R3,-(SP) ; SAVE R3
79 MOV #FNC,R3 ; GET FUNCTION CODE
80 .IF B <ARG1>
81 BR DDCM1 ; COMMON PROCESSING FOR NON-CCB CONTROL FUNCTION
82 .IFF
83 .IF IDN <ARG1>,<MDC>
84 BR DDCM2 ; COMMON PROCESSING FOR MODEM CONTROL FUNCTIONS
85 .IFF
86 .IF IDN <ARG1>,<SUB>
87 BR DDCM4
88 .IFF
89 BR DDCM3 ; COMMON PROCESSING FOR CONTROL FUNCTIONS WITH CCB
90 .ENDC
91 .ENDC
92 .ENDC
93 .ENDM DDFDF
94 ;
95 .IFF ; DF K$$DAS
96 ;
97 .MACRO DDFDF,FNC,ARG1
98 .IF B <ARG1>
99 JSR R3,DDCM1 ; COMMON PROCESSING FOR NON-CCB CONTROL FUNCTION
100 .IFF
101 .IF IDN <ARG1>,<MDC>
102 JSR R3,DDCM2 ; COMMON PROCESSING FOR MODEM CONTROL FUNCTIONS
103 .IFF
104 .IF IDN <ARG1>,<SUB>
105 JSR R3,DDCM4
106 .IFF
107 JSR R3,DDCM3 ; COMMON PROCESSING FOR CONTROL FUNCTIONS WITH CCB
108 .ENDC
109 .ENDC
110 .ENDC
111 10$: .WORD FNC
112 .ENDM
113 .ENDC ; DF K$$DAS
114

```

|                  |                  |                   |                  |                  |
|------------------|------------------|-------------------|------------------|------------------|
| A\$\$CHK= 000000 | CS.ENB= 000020   | D\$\$SYNM= 000000 | I\$\$RDN= 000000 | M\$\$FCS= 000000 |
| A\$\$CPS= 000000 | CS.ERR= 100000   | E\$\$XPR= 000000  | K\$\$CNT= 177546 | M\$\$MGE= 000000 |
| A\$\$PRI= 000000 | CS.FTL= 001000   | FC.CCP= 000020    | K\$\$CSR= 177546 | M\$\$NET= 000000 |
| A\$\$TRP= 000000 | CS.HCR= 000001   | FC.CTL= 000006    | K\$\$LDC= 000000 | M\$\$OVR= 000000 |
| CB.CCB= 000002   | CS.HFE= 002000   | FC.KCP= 000016    | K\$\$TPS= 000074 | N\$\$ACC= 000001 |
| CB.DDM= 000040   | CS.LST= 040000   | FC.KIL= 000004    | LD\$LP= 000000   | N\$\$BUF= 000001 |
| CB.DLC= 000020   | CS.MTL= 004000   | FC.MAN= 000024    | LF.ACT= 100000   | N\$\$LDV= 000001 |
| CB.RDS= 000004   | CS.RNG= 000010   | FC.MLD= 000026    | LF.BRO= 000400   | N\$\$MCP= 000001 |
| CB.SDB= 000010   | CS.ROV= 000004   | FC.PCT= 000030    | LF.BWT= 000007   | N\$\$MLL= 000001 |
| CB.SLI= 000100   | CS.RSN= 010000   | FC.PWR= 000022    | LF.ENA= 002000   | N\$\$MOV= 000010 |
| CB.XLB= 000001   | CS.SHU= 000001   | FC.RCE= 000002    | LF.LPB= 001000   | N\$\$NCT= 000001 |
| CC.LLC= 000200   | CS.SID= 000002   | FC.RCP= 000014    | LF.MDC= 000100   | N\$\$PEM= 000001 |
| CE.ABO= 100362   | CS.SUR= 000004   | FC.TJM= 000010    | LF.MFL= 004000   | PR7 = ***** GX   |
| CE.DAO= 100346   | CS.SUC= 000001   | FC.XCP= 000012    | LF.MTP= 000020   | PS = *****       |
| CE.DIS= 100366   | CS.TMO= 020000   | FC.XME= 000000    | LF.PAC= 000200   | P\$\$P45= 000000 |
| CE.ERR= 100370   | CS.XUR= 000004   | FS.AST= 000000    | LF.RDY= 040000   | P\$\$WRD= 000000 |
| CE.ILN= 100350   | C\$\$CKP= 000000 | FS.CIB= 002000    | LF.REA= 010000   | Q\$\$OPT= 000010 |
| CE.LTD= 100356   | C\$\$ORE= 000400 | FS.CRA= 001000    | LF.SER= 000040   | R\$\$DER= 000000 |
| CE.MOP= 100372   | C\$\$RSR= 177564 | FS.DIS= 013000    | LF.TIM= 000010   | R\$\$K1I= 000001 |
| CE.NTE= 100361   | C.ADD= 000034    | FS.DVC= 001000    | LF.UNL= 020000   | R\$\$SND= 000000 |
| CE.RTE= 100376   | C.BID= 000003    | FS.ENB= 012000    | LF.X2P= 000000   | R\$\$11M= 000000 |
| CE.SRC= 100364   | C.BUF= 000014    | FS.EXI= 001000    | LN.CLO= 000000   | SF.ACT= 000200   |
| CE.STP= 100352   | C.BUF1= 000014   | FS.GET= 006000    | LN.DUM= 000005   | SF.ENA= 000100   |
| CE.TME= 100354   | C.BUF2= 000024   | FS.HLT= 000000    | LN.LOA= 000004   | SF.LPB= 000004   |
| CE.TMO= 100374   | C.CNT= 000020    | FS.INI= 000000    | LN.LOO= 000003   | SF.MFL= 000040   |
| CE.UNS= 100344   | C.CNT1= 000020   | FS.KIL= 000000    | LN.OAU= 000003   | SF.PAC= 000020   |
| CF.CHN= 000001   | C.CNT2= 000030   | FS.LCL= 100000    | LN.OFF= 000001   | SF.REA= 000010   |
| CF.EOM= 000004   | C.FLG= 000022    | FS.LTM= 001000    | LN.ON= 000000    | SF.SER= 000001   |
| CF.HUR= 000020   | C.FLG1= 000022   | FS.MNT= 004000    | LN.OOP= 000004   | SF.SVC= 000002   |
| CF.LB= 100000    | C.FLG2= 000032   | FS.MSN= 014000    | LN.OPE= 000001   | SF.UNL= 000040   |
| CF.LIN= 000002   | C.FNC= 000010    | FS.REA= 001000    | LN.REF= 000002   | S\$\$WRG= 000000 |
| CF.SOM= 000010   | C.LIN= 000006    | FS.RET= 000000    | LN.SER= 000002   | S\$\$YSZ= 007600 |
| CF.SYN= 000040   | C.LNK= 000000    | FS.REZ= 003000    | LN.STA= 000017   | S.COST= 000001   |
| CF.TRN= 000100   | C.MOD= 000011    | FS.RLB= 002000    | LN.SUB= 000360   | S.FLG= 000000    |
| CM.CIR= 000002   | C.NSP= 000004    | FS.RNG= 011000    | LN.TRI= 000006   | S.LEN= 000004    |
| CM.FMT= 100000   | C.PRO= 000042    | FS.RST= 000000    | L\$\$ASG= 000000 | S.NMST= 000002   |
| CM.HRD= 000002   | C.RSV= 000002    | FS.RTN= 001000    | L\$\$DRV= 000000 | S.OWNR= 000003   |
| CM.LIN= 000000   | C.STA= 000007    | FS.SET= 005000    | L\$\$P11= 000001 | T\$\$KMG= 000000 |
| CM.LOD= 000001   | C.STS= 000012    | FS.SFC= 005000    | L\$\$11R= 000000 | T\$\$MIN= 000000 |
| CM.XLO= 000004   | C.URM= 177776    | FS.SFR= 006000    | L.COST= 000015   | V\$\$CTR= 001000 |
| CP.DCF= 000040   | C.XACP= 000004   | FS.SFS= 004000    | L.CTL= 000012    | X\$\$DBT= 000000 |
| CP.HDL= 000007   | C.XID= 000035    | FS.SPW= 040000    | L.CVA= 177776    | X\$\$MDC= 000001 |
| CP.PS= 177400    | C.XLEN= 000044   | FS.STM= 000000    | L.DDM= 000002    | ZF.COU= 001000   |
| CP.PSI= 000200   | C.XPLI= 000040   | FS.STP= 002000    | L.DDS= 000004    | ZF.DDM= 000001   |
| CP.XCF= 000100   | C.XPT= 000034    | FS.STR= 001000    | L.DLC= 000003    | ZF.DIA= 004000   |
| CP.2FR= 000030   | C.XSVC= 000042   | FS.TRM= 003000    | L.DLM= 000006    | ZF.DLC= 000002   |
| CS.ABO= 000100   | C.XTC= 000037    | FS.WLB= 001000    | L.DLS= 000010    | ZF.DVP= 100000   |
| CS.BRO= 000002   | C.X25= 000036    | FS.XKL= 002000    | L.FLG= 000000    | ZF.INI= 040000   |
| CS.BUF= 000200   | DDCM1= 000140R   | FS.XOF= 010000    | L.KRBA= 000016   | ZF.KMX= 000020   |
| CS.CES= 000002   | DDCM2= 000150R   | FS.XON= 007000    | L.LEN= 000022    | ZF.LLC= 000004   |
| CS.CHN= 000010   | DDCM3= 000254R   | FS.ZER= 002000    | L.MPF= 000022    | ZF.LMC= 000100   |
| CS.CMP= 000200   | DDCM4= 000262R   | F\$\$LVL= 000001  | L.NMST= 000020   | ZF.MAN= 020000   |
| CS.DCR= 000400   | DDMDSP= 000312R  | G\$\$TPP= 000000  | L.NSTA= 000014   | ZF.MFL= 000010   |
| CS.DEF= 000004   | D\$\$BUG= 177514 | G\$\$TSS= 000000  | L.OWNR= 000021   | ZF.MTM= 000400   |
| CS.DEV= 000002   | D\$\$ISK= 000000 | G\$\$TIK= 000000  | L.UNT= 000013    | ZF.MUX= 000040   |
| CS.DIS= 000040   | D\$\$L11= 000001 | G\$\$WRD= 000000  | M\$\$CRB= 000124 | ZF.PSE= 002000   |
| CS.ENA= 000001   | D\$\$YNC= 000000 | I\$\$RAR= 000000  | M\$\$CRX= 000000 | ZF.SLI= 010000   |

|     |                |                   |                             |
|-----|----------------|-------------------|-----------------------------|
| 174 | \$DDXME::DDFDF | FC.XME,SUB        | ; TRANSMIT ENABLE           |
| 175 | \$DDRCE::DDFDF | FC.RCE,SUB        | ; RECEIVE ENABLE            |
| 176 |                |                   |                             |
| 177 | \$DDMAN::DDFDF | FC.MAN,SUB        | ; NETWORK MANAGEMENT        |
| 178 |                |                   |                             |
| 179 | \$DDXKL::DDFDF | FC.KIL+FS.XKL,CCB | ; TRANSMIT KILL             |
| 180 | \$DDCRA::DDFDF | FC.KIL+FS.CRA,CCB | ; RECEIVE KILL              |
| 181 | \$DDKIL::DDFDF | FC.KIL+FS.KIL,CCB | ; RECEIVE AND TRANSMIT KILL |
| 182 | \$DDSTR::DDFDF | FC.CTL+FS.STR,CCB | ; START                     |
| 183 | \$DDSTP::DDFDF | FC.CTL+FS.STP,CCB | ; STOP                      |
| 184 | \$DDGET::DDFDF | FC.CTL+FS.GET,CCB | ; GET CHARACTERISTICS       |
| 185 | \$DDSET::DDFDF | FC.CTL+FS.SET,CCB | ; SET CHARACTERISTICS       |
| 186 |                |                   |                             |
| 187 | \$DDRNG::DDFDF | FC.CTL+FS.RNG,MDC | ; LOOK FOR RING             |
| 188 | \$DDENB::DDFDF | FC.CTL+FS.ENB,MDC | ; ENABLE LINE               |
| 189 | \$DDDIS::DDFDF | FC.CTL+FS.DIS,MDC | ; DISABLE LINE              |
| 190 |                |                   |                             |
| 191 | \$DDXON::DDFDF | FC.CTL+FS.XON     | ; XON                       |
| 192 | \$DDXOF::DDFDF | FC.CTL+FS.XOF     | ; XOFF                      |
| 193 | \$DDMSN::DDFDF | FC.CTL+FS.MSN     | ; SENSE MODEM STATUS        |
| 194 |                |                   |                             |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE    | REFERENCES    |
|---------|----------|---------------|
| C.LIN   | 000006   | 15-560        |
| I\$SAS  | = *****  | 5-67          |
| K\$SDAS | = *****  | 6-75          |
| LF.ACT  | = 100000 | #5-68         |
| LF.BRO  | = 000400 | #5-68         |
| LF.BWT  | = 000007 | #5-68         |
| LF.ENA  | = 002000 | #5-68         |
| LF.LPB  | = 001000 | #5-68         |
| LF.MDC  | = 000100 | #5-68         |
| LF.MFL  | = 004000 | #5-68         |
| LF.MTP  | = 000020 | #5-68         |
| LF.PAC  | = 000200 | #5-68         |
| LF.RDY  | = 040000 | #5-68         |
| LF.REA  | = 010000 | #5-68         |
| LF.SER  | = 000040 | #5-68         |
| LF.TIM  | = 000010 | #5-68         |
| LF.UNL  | = 020000 | #5-68         |
| LF.X2P  | = 000000 | #5-68         |
| LN.CLO  | = 000000 | #5-68         |
| LN.DUM  | = 000005 | #5-68         |
| LN.LOA  | = 000004 | #5-68         |
| LN.LOO  | = 000003 | #5-68         |
| LN.OAU  | = 000003 | #5-68         |
| LN.OFF  | = 000001 | #5-68         |
| LN.ON   | = 000000 | #5-68         |
| LN.OOP  | = 000004 | #5-68         |
| LN.OPE  | = 000001 | #5-68         |
| LN.REF  | = 000002 | #5-68         |
| LN.SER  | = 000002 | #5-68         |
| LN.STA  | = 000017 | #5-68         |
| LN.SUB  | = 000360 | #5-68         |
| LN.TRI  | = 000006 | #5-68         |
| L.COST  | 000015   | #5-68         |
| L.CTL   | 000012   | #5-68         |
| L.CVA   | 177776   | #5-68         |
| L.DDM   | 000002   | #5-68         |
| L.DDS   | 000004   | #5-68         |
| L.DLC   | 000003   | #5-68         |
| L.DLM   | 000006   | #5-68         |
| L.DLS   | 000010   | #5-68         |
| L.FLG   | 000000   | #5-68         |
| L.KRBA  | = 000016 | #5-68         |
| L.LEN   | = 000022 | #5-68         |
| L.MPF   | 000022   | #5-68         |
| L.NMST  | 000020   | #5-68         |
| L.NSTA  | 000014   | #5-68         |
| L.OWNR  | 000021   | #5-68         |
| L.UNT   | 000013   | #5-68         |
| N\$1LN  | = *****  | 15-558 15-564 |
| R\$1TD  | = *****  | 5-67          |
| R\$11M  | = 000000 | 5-67          |
| R\$11S  | = *****  | 5-67          |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL | VALUE    | REFERENCES    |
|--------|----------|---------------|
| C.BID  | 000003   | *7-190        |
| C.CNC  | 000010   | 5-111         |
| C.LIN  | 000006   | *6-142 *7-175 |
| C.STA  | 000007   | *6-143 7-187  |
| C.STS  | 000012   | *7-189        |
| FC.CCP | = 000020 | 6-142 7-178   |
| FC.KCP | = 000016 | 7-181         |
| FC.RCP | = 000014 | 7-184         |
| FC.XCP | = 000012 | 7-175         |
| FS.AST | = 000000 | 6-142         |
| ISBAS  | = *****  | 4-63          |
| LF.ACT | = 100000 | #4-64         |
| LF.BRO | = 000400 | #4-64         |
| LF.BWT | = 000007 | #4-64         |
| LF.ENA | = 002000 | #4-64         |
| LF.LPB | = 001000 | #4-64         |
| LF.MDC | = 000100 | #4-64         |
| LF.MFL | = 004000 | #4-64         |
| LF.MTP | = 000020 | #4-64         |
| LF.PAC | = 000200 | #4-64         |
| LF.RDY | = 040000 | #4-64         |
| LF.REA | = 010000 | #4-64         |
| LF.SER | = 000040 | #4-64         |
| LF.TIM | = 000010 | #4-64         |
| LF.UNL | = 020000 | #4-64         |
| LF.X2P | = 000000 | #4-64         |
| LN.CLO | = 000000 | #4-64         |
| LN.DJM | = 000005 | #4-64         |
| LN.LOA | = 000004 | #4-64         |
| LN.LOO | = 000003 | #4-64         |
| LN.OAU | = 000003 | #4-64         |
| LN.OFF | = 000001 | #4-64         |
| LN.ON  | = 000000 | #4-64         |
| LN.OOP | = 000004 | #4-64         |
| LN.OPE | = 000001 | #4-64         |
| LN.REF | = 000002 | #4-64         |
| LN.SER | = 000002 | #4-64         |
| LN.STA | = 000017 | #4-64         |
| LN.SUB | = 000360 | #4-64         |
| LN.TRI | = 000006 | #4-64         |
| L.COST | 000015   | #4-64         |
| L.CTL  | 000012   | #4-64         |
| L.CVA  | 177776   | #4-64         |
| L.DDM  | 000002   | #4-64         |
| L.DDS  | 000004   | #4-64         |
| L.DLC  | 000003   | #4-64 5-109   |
| L.DLM  | 000006   | #4-64         |
| L.DLS  | 000010   | #4-64 5-108   |
| L.FLG  | 000000   | #4-64         |
| L.KRBA | 000016   | #4-64         |
| L.LEN  | = 000022 | #4-64         |
| L.MPF  | 000022   | #4-64         |

CELLC MACRO V05.03b Friday 28-Jun-85 18:19 Page 7-2  
Symbol table

|       |          |          |          |          |          |          |          |            |          |
|-------|----------|----------|----------|----------|----------|----------|----------|------------|----------|
| Z.FLG | 000010   | Z.NAM    | 000004   | \$DL CRS | 000074RG | \$LL CRS | 000074RG | \$PDSPL=   | ***** GX |
| Z.LEN | = 000016 | Z.PCB    | 000012   | \$LL CAL | 000146RG | \$NMC RS | 000074RG | \$PDVTA=   | ***** GX |
| Z.LLN | 000006   | Z.SCH    | 000007   | \$LL CLC | 000126RG | \$PDDSP= | ***** GX | \$STDLC=   | ***** GX |
| Z.MAP | 000020   | \$CMPDV= | ***** GX | \$LL CRQ | 000000RG | \$PDQU1= | ***** GX | .\$\$\$\$= | 000034   |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000174 001 (RW,I,LCL,REL,CON)

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 7  
Work file writes: 9  
Size of work file: 17290 Words ( 68 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:13.15

SY:CELLC.V2,[130,134]CELLC/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]CELLC



\*\*FILE\*\*ID\*\*CELOG1

```

CCCCCCCC EEEEEEEEE LL 000000 GGGGGGGG 11
CCCCCCCC EEEEEEEEE LL 000000 GGGGGGGG 11
CC EE LL 00 00 GG 1111
CC EE LL 00 00 GG 1111
CC EE LL 00 00 GG 11
CC EE LL 00 00 GG 11
CC EEEEEEEE LL 00 00 GG 11
CC EEEEEEEE LL 00 00 GG 11
CC EE LL 00 00 GG 11
CC EE LL 00 00 GG 11
CC EE LL 00 00 GG 11
CC EE LL 00 00 GG 11
CC EE LL 00 00 GG 11
CC EE LL 00 00 GG 11
CC EE LL 00 00 GG 11
CCCCCCCC EEEEEEEEE LLLLLLLLL 000000 GGGGGG 111111
CCCCCCCC EEEEEEEEE LLLLLLLLL 000000 GGGGGG 111111

```

```

....
....
....
....

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```

```

131
132 000020 016667 000004 000000G MOV 4(SP),KISAR5 ;; Restore previous APR contents
133 ;; Current contents of stack:
134 ;; R4 (Saved by JSR R4,$INTSX in DDM)
135 ;; R5 (Saved by JSR R5,'ISR' in Line Table)
136 ;; KISAR5 (Saved in Line Table)
137 ;; PS (Saved by Interrupt)
138 ;; PC (Saved by Interrupt)
139 000026 016666 000002 000004 MOV 2(SP),4(SP) ;; Move saved R5 down on stack
140 000034 012616 MOV (SP)+,(SP) ;; Move saved R4 down on stack
141 ;; Contents of stack (necessary for $INTXX)
142 ;; R4 (Saved by JSR R4,$INTSX in DDM)
143 ;; R5 (Saved by JSR R5,'ISR' in Line Table)
144 ;; PS (Saved by Interrupt)
145 ;; PC (Saved by Interrupt)
146 .ENDC ;; DF K$$DAS
147
148 000036 000167 000000G JMP $INTX1 ;; Return through RSX Interrupt exit ;sjp0005

```

\*\*FILE\*\*ID\*\*CESUB

E 15

```
CCCCCCCC EEEEEEEEE SSSSSSSS UU UU BBBB8888
CCCCCCCC EEEEEEEEE SSSSSSSS UU UU BBBB8888
CC EE SS UU BB BB
CC EE SS UU BB BB
CC EE SS UU BB BB
CC EEEEEEE SSSSSS UU UU BBBB8888
CC EEEEEEE SSSSSS UU UU BBBB8888
CC EE SS UU BB BB
CC EE SS UU BB BB
CC EE SS UU BB BB
CC EE SS UU BB BB
CCCCCCCC EEEEEEEEE SSSSSSSS UUUUUUUUU BBBB8888
CCCCCCCC EEEEEEEEE SSSSSSSS UUUUUUUUU BBBB8888
....
....
....
....
```

```
LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LLLLLLLLL SSSSSSSS TT
LLLLLLLLL SSSSSSSS TT
```

F 15

```

450 .SBTTL $CEMUL - UNSIGNED MULTIPLY
451
452 ;+
453 ;*- $CEMUL - UNSIGNED MULTIPLY
454 ; THIS SUBROUTINE PERFORMS UNSIGNED MULTIPLICATION WHERE THE RESULT IS LESS
455 ; THAN 16 BITS. IT IS INCLUDED HERE SO THAT WE CAN CONDITIONALISE THE CODE
456 ; FOR EXTENDED INSTRUCTION SET.
457
458 ; NOTE THAT IT IS THE CALLERS RESPONSIBILITY TO ENSURE THAT THE RESULT
459 ; WILL FIT IN 16 BITS.
460
461 ; INPUTS:
462 ; R0 = MULTIPLIER
463 ; R1 = MULTIPLICAND
464
465 ; OUTPUTS:
466 ; R0 = CORRUPTED
467 ; R1 = LOW 16 BITS OF RESULT
468 ; -
469
470 000572 $CEMUL:::IF DF R$$EIS
471
472 MUL R0,R1 ; PERFORM MULTIPLICATION
473 RETURN
474
475 .IFF ; DF R$$EIS
476
477 000572 CALLR $MUL ; CALL EXEC MULTIPLY ROUTINE
478
479 .ENDC ; DF R$$EIS

```

Define Stack Offsets and Common Buffer Return Table

```

56 .SBTTL Define Stack Offsets and Common Buffer Return Table
57
58 ;
59 .MCALL INHIB$,ENABL$,SAVRG,RESRG,SAVMAP,RESMAP,MAP
60 .MCALL CCBDF$,SLTDF$
61 ;
62 CCBDF$; Define CCB Offsets
63 SLTDF$; Define SLT Offsets
64 ;
65 ; Define offsets to saved PSW from $INHIB. This offset is used by the
66 ; FAIL entry (refer to BUFGT) to set the C-Bit so that failures to allocate
67 ; an RDB, LDB, or SDB will set the C-Bit on the ENABL$.
68 000000 PRIOFF=0 ; Offset to Saved Priority
69
70 .IF DF M$$PRO
71 PRIOFF=2
72 .ENDC ; Extra word on Stack for Multi-Processors
73
74

```

\$CCBGT - Allocate a Standard CCB

+  
.  
.  
.  
.  
.  
.  
.  
.  
.  
.  
.  
.  
.  
-

Allocation Control Blocks are used to control the allocation of fixed length blocks from doubly link lists (address doublewords). The control blocks have the following format:

```
.WORD <# of buffers available>
.WORD 0,0 ; Allocation listhead
.WORD <# of allocation failures>
```

```

56 .SBTTL Define Stack Offsets and Common Buffer Return Table
57
58 ;
59 .MCALL INHIB$,ENABL$,SAVRG,RESRG,SAVMAP,RESMAP,MAP
60 .MCALL CCBDF$,SLTDF$
61 ;
62 000000 CCBDF$; Define CCB Offsets
63 000000 SLTDF$; Define SLT Offsets
64
65 ; Define offsets to saved PSW from $INHIO. This offset is used by the
66 ; FAIL entry (refer to BUFGT) to set the C-Bit so that failures to allocate
67 ; an RDB, LDB, or SDB will set the C-Bit on the ENABL$.
68
69 000000 PRIOFF=0 ; Offset to Saved Priority
70
71 .IF DF M$$PRO
72 PRIOFF=2 ; Extra word on Stack for Multi-Processors
73 .ENDC
74

```

537  
538  
539  
540  
541  
542  
543  
544  
545  
546

:+  
:  
:  
:  
:  
:  
:  
:  
:-

# .SBTTL Allocation Control Block Documentation

Allocation Control Blocks are used to control the allocation of fixed length blocks from doubley link lists (address doublewords). The control blocks have the following format:

.WORD <# of buffers available>  
.WORD 0,0 ; Allocation listhead  
.WORD <# of allocation failures>



```
58
59
60
61
62 ; MACRO LIBRARY CALLS
63 ;
64 .MCALL INHIB$,ENABL$,SAVRG,RESRG
65 .MCALL CCBDF$,PDVDF$,SLTDF$
66 .MCALL CALLR ; AVOID SYSTEM DEPENDENCY
67 CCBDF$; DEFINE THE CCB OFFSETS
68 PDVDF$; DEFINE THE PDV OFFSETS
69 SLTDF$; DEFINE THE SLT OFFSETS
```

```

533 .SBTTL $STDDM - SET DDM PDV INDEX AND LINE TABLE ADDRESS
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558 000404 $STDDM::IF NDF N$$1LN
559
560 000404 116402 000006 MOVB C.LIN(R4),R2 ; EXTRACT SYSTEM LINE NUMBER
561
562 .ENDC
563
564 000410 $STDD1::IF DF N$$1LN
565
566 MOV @$$SLTMA,R2 ; GET ADDRESS OF SYSTEM LINE ENTRY
567
568 .IFF
569
570 000410 006302 ASL R2 ; FORM WORD INDEX
571 000412 066702 000000G ADD $$SLTMA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE
572 000416 011202 MOV (R2),R2 ; GET ADDRESS OF SYSTEM LINE ENTRY
573
574 .ENDC
575
576 000420 016205 000004 MOV L.DDS(R2),R5 ; GET DEVICE LINE TABLE ADDRESS
577 000424 116202 000002 MOVB L.LDM(R2),R2 ; GET DEVICE DRIVER PDV INDEX (WORD INDEX)
578 000430 RETURN ; RETURN
579
580
581 .END

```

```

116 .IF NDF X$$NDM
117 .SBTTL $DD??? - DLC TO DDM REQUESTS
118
119 +
120 GENERAL REQUESTS WITH CCB
121
122 :
123 : FORMAT OF CALL:
124 : CALL $DD???
125
126 : INPUTS:
127 : R4 = ADDRESS OF FIRST CCB IN CHAIN
128 : (ALL CCBs MUST CONTAIN A VALID SLN)
129
130 : OUTPUTS TO DDM:
131 : R4 = ADDRESS OF FIRST CCB IN CHAIN
132 : R5 = ADDRESS OF DDM LINE TABLE
133 : R2 & R3 - AVAILABLE FOR USE WITHOUT SAVING
134
135 : REGISTERS ACROSS CALL:
136 : R0,R1 - MUST BE PRESERVED BY DDM IF USED
137 : R2,R3,R5 - PRESERVED BY COMM EXEC
138 : R4 - MAY BE MODIFIED
139
140 : ON RETURN TO DLC:
141 : C-BIT CLEAR - REQUEST HAS COMPLETED SYNCHRONOUSLY
142 : R4 = ADDRESS OF CCB
143
144 : C-BIT SET - REQUEST WILL COMPLETE ASYNCHRONOUSLY
145 : R4 = 0
146 :
147 :
148 +
149 SPECIAL REQUESTS WITHOUT CCB
150
151 :
152 : FORMAT OF CALL:
153 : CALL $DD???
154
155 : INPUTS:
156 : R3 = SYSTEM LINE NUMBER
157 : R4 = OPTIONAL CALLING PARAMETER TO DDM
158
159 : OUTPUTS TO DDM:
160 : R4 = OPTIONAL CALLING PARAMETER FROM DLC
161 : R5 = ADDRESS OF DDM LINE TABLE
162 : R2 & R3 - AVAILABLE FOR USE WITHOUT SAVING
163
164 : REGISTERS ACROSS CALL:
165 : R0,R1 - MUST BE PRESERVED BY DDM IF USED
166 : R2,R3,R5 - PRESERVED BY COMM EXEC
167 : R4 - MAY BE MODIFIED
168
169 : ON RETURN TO DLC:
170 : C-BIT ALWAYS CLEAR
171 : R4 = OPTIONAL RETURNING PARAMETER FROM DDM
172 :

```

|                |                   |                  |                  |                    |
|----------------|-------------------|------------------|------------------|--------------------|
| ZF.TIM= 000200 | Z.MAP 000020      | \$DDDIS 000110RG | \$DDRNG 000074RG | \$DDXON 000116RG   |
| ZF.X3P= 000000 | Z.NAM 000004      | \$DDENB 000102RG | \$DDSET 000066RG | \$LLCSP 000500RG   |
| ZS.ASN= 100000 | Z.PCB 000012      | \$DDGET 000060RG | \$DDSPC 000500RG | \$PDDSP= ***** GX  |
| ZS.BSY= 140000 | Z.SCH 000007      | \$DDKCP 000450RG | \$DDSTP 000052RG | \$PDQU1= ***** GX  |
| Z.AVL 000014   | \$CCBGT= ***** GX | \$DDKIL 000036RG | \$DDSTR 000044RG | \$PDVTA= ***** GX  |
| Z.DAT 000016   | \$CCBRT= ***** GX | \$DDMAN 000014RG | \$DDXKL 000022RG | \$SLTMA= ***** GX  |
| Z.DSP 000000   | \$DDAST 000400RG  | \$DDMSN 000132RG | \$DDXME 000000RG | \$STDDM 000510RG   |
| Z.FLG 000010   | \$DDCCP 000460RG  | \$DDRCE 000006RG | \$DDXMP 000430RG | \$STDD1 000514RG   |
| Z.LEN = 000016 | \$DDCRA 000030RG  | \$DDRCP 000440RG | \$DDXOF 000124RG | .\$\$\$\$ = 000034 |
| Z.LLN 000006   |                   |                  |                  |                    |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000536 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 9  
Work file writes: 12  
Size of work file: 17461 Words ( 69 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:19.49  
SY: CEDMM.V2,[130,134]CEDMM/CH/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]V2.CEDMM

196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234

.SBTIL DDCM1 - COMMON PROCESS FOR SPECIAL NON-CCB FUNCTIONS

:+  
:\*\*--DDCM1 - COMMON PROCESS FOR SPECIAL NON-CCB FUNCTIONS

INPUTS:

R3 = ADDRESS OF FUNCTION CODE  
R4 = UNSPECIFIED CALLING PARAMETER TO DDM  
STACK CONTAINS:  
00(SP) = LINE NUMBER (ORIGINALLY IN R3)  
02(SP) = RETURN ADDRESS TO CALLING DLC

OUTPUTS TO DDM:

R3 = SUBFUNCTION CODE (WORD INDEX)  
R4 = SAME AS ON ENTRY  
R5 = LINE TABLE ADDRESS

OUTPUTS ON RETURN TO CALLER:

FUNCTION HAS BEEN PERFORMED  
R4 = UNSPECIFIED RETURN PARAMETER TO DLC FROM DDM

REGISTERS ACCROSS CALL:

R2,R3,R5 PRESERVED  
R4 MAY BE MODIFIED BY DDM

DDCM1: .ENABL LSB  
.IF DF K\$\$DAS

MOV R3,\$DDFNC ; STORE FUNCTION CODE IN DATA AREA  
MOV #\$\$DFNC,R3 ; POINT R3 AT FUNCTION CODE

.ENDC

MOV R2,-(SP) ; SAVE R2  
MOV 2(SP),R2 ; GET LINE NUMBER  
BR 22\$ ; JOIN COMMON CODE TO DISPATCH TO DDM

CEDDMN      CREATED BY    MACRO    ON 28-JUN-85 AT 18:18      PAGE 2      F 10

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE         | REFERENCES              |
|---------|---------------|-------------------------|
| SF.ACT  | = 000200      | #5-68                   |
| SF.ENA  | = 000100      | #5-68                   |
| SF.LPB  | = 000004      | #5-68                   |
| SF.MFL  | = 000040      | #5-68                   |
| SF.PAC  | = 000020      | #5-68                   |
| SF.REA  | = 000010      | #5-68                   |
| SF.SER  | = 000001      | #5-68                   |
| SF.SVC  | = 000002      | #5-68                   |
| SF.UNL  | = 000040      | #5-68                   |
| S.COST  | 000001        | #5-68                   |
| S.FLG   | 000000        | #5-68                   |
| S.LEN   | 000004        | #5-68                   |
| S.NMST  | 000002        | #5-68                   |
| S.OWNR  | 000003        | #5-68                   |
| X\$MBCB | = *****       | 5-67                    |
| X\$MDC  | = *****       | 5-2                     |
| X\$NDM  | = 000001      | 5-3                     |
| ZF.COU  | = 001000      | 5-1      5-3      7-116 |
| ZF.DDM  | = 000001      | #5-67                   |
| ZF.DIA  | = 004000      | #5-67                   |
| ZF.DLC  | = 000002      | #5-67                   |
| ZF.DVP  | = 100000      | #5-67                   |
| ZF.INI  | = 040000      | #5-67                   |
| ZF.KMX  | = 000020      | #5-67                   |
| ZF.LLC  | = 000004      | #5-67                   |
| ZF.LMC  | = 000100      | #5-67                   |
| ZF.MAN  | = 020000      | #5-67                   |
| ZF.MFL  | = 000010      | #5-67                   |
| ZF.MTM  | = 000400      | #5-67                   |
| ZF.MUX  | = 000040      | #5-67                   |
| ZF.PSE  | = 002000      | #5-67                   |
| ZF.SLI  | = 010000      | #5-67                   |
| ZF.TIM  | = 000200      | #5-67                   |
| ZF.X3P  | = 000000      | #5-67                   |
| ZS.ASN  | = 100000      | #5-67                   |
| ZS.BSY  | = 140000      | #5-67                   |
| Z.AVL   | 000014        | #5-67                   |
| Z.DAT   | 000016        | #5-67                   |
| Z.DSP   | 000000        | 5-67                    |
| Z.FLG   | 000010        | #5-67                   |
| Z.LEN   | = 000016      | #5-67                   |
| Z.LLN   | 000006        | #5-67                   |
| Z.MAP   | 000020        | #5-67                   |
| Z.NAM   | 000004        | #5-67                   |
| Z.PCB   | 000012        | #5-67                   |
| Z.SCH   | = 000007      | #5-67                   |
| \$SLTMA | = *****    GX | 15-571                  |
| \$STDDM | 000000    RG  | #15-558                 |
| \$STDD1 | 000004    RG  | #15-564                 |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL | VALUE    | REFERENCES |
|--------|----------|------------|
| L.NMST | 000020   | #4-64      |
| L.NSTA | 000014   | #4-64      |
| L.OWNR | 000021   | #4-64      |
| L.UNT  | 000013   | #4-64      |
| M\$PRO | = *****  | 7-202      |
| N\$1LN | = *****  | 5-90       |
| R\$11D | = *****  | 4-63       |
| R\$11M | = 000000 | 4-63       |
| R\$11S | = *****  | 4-63       |
| SF.ACT | = 000200 | #4-64      |
| SF.ENA | = 000100 | #4-64      |
| SF.LPB | = 000004 | #4-64      |
| SF.MFI | = 000040 | #4-64      |
| SF.PAC | = 000020 | #4-64      |
| SF.REA | = 000010 | #4-64      |
| SF.SER | = 000001 | #4-64      |
| SF.SVC | = 000002 | #4-64      |
| SF.UNL | = 000040 | #4-64      |
| S.COST | = 000001 | #4-64      |
| S.FLG  | = 000000 | #4-64      |
| S.LEN  | = 000004 | #4-64      |
| S.NMST | = 000002 | #4-64      |
| S.OWNR | = 000003 | #4-64      |
| X\$MCB | = *****  | 4-63       |
| ZF.COU | = 001000 | #4-63      |
| ZF.DDM | = 000001 | #4-63      |
| ZF.DIA | = 004000 | #4-63      |
| ZF.DLC | = 000002 | #4-63      |
| ZF.DVP | = 100000 | #4-63      |
| ZF.INI | = 040000 | #4-63      |
| ZF.KMX | = 000020 | #4-63      |
| ZF.LLC | = 000004 | #4-63      |
| ZF.LMC | = 000100 | #4-63      |
| ZF.MAN | = 020000 | #4-63      |
| ZF.MFL | = 000010 | #4-63      |
| ZF.MTM | = 000400 | #4-63      |
| ZF.MUX | = 000040 | #4-63      |
| ZF.PSE | = 002000 | #4-63      |
| ZF.SLI | = 010000 | #4-63      |
| ZF.TIM | = 000200 | #4-63      |
| ZF.X3P | = 000000 | #4-63      |
| ZS.ASN | = 100000 | #4-63      |
| ZS.BSY | = 140000 | #4-63      |
| Z.AVL  | = 000014 | #4-63      |
| Z.DAT  | = 000016 | #4-63      |
| Z.DSP  | = 000000 | #4-63      |
| Z.FLG  | = 000010 | #4-63      |
| Z.LEN  | = 000016 | #4-63      |
| Z.LLN  | = 000006 | #4-63      |
| Z.MAP  | = 000020 | #4-63      |
| Z.NAM  | = 000004 | #4-63      |
| Z.PCB  | = 000012 | #4-63      |

CELLC CREATED BY MACRO ON 28-JUN-85 AT 18:19 PAGE 1 F 12  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL | VALUE      | REFERENCES          |
|--------|------------|---------------------|
| C.BID  | 000003     | *6-176              |
| C.FNC  | 000010     | 7-217               |
| C.LIN  | 000006     | 5-114 *5-117 *6-175 |
| C.STA  | 000007     | 7-218 *7-219        |
| ISSAS  | = *****    | 4-61                |
| LF.ACT | = 100000   | #4-62               |
| LF.BRO | = 000400   | #4-62               |
| LF.BWT | = 000007   | #4-62               |
| LF.ENA | = 002000   | #4-62               |
| LF.LPB | = 001000   | #4-62               |
| LF.MDC | = 000100   | #4-62               |
| LF.MFL | = 004000   | #4-62               |
| LF.MTP | = 000020   | #4-62               |
| LF.PAC | = 000200   | #4-62               |
| LF.RDY | = 040000   | #4-62               |
| LF.REA | = 010000   | #4-62               |
| LF.SER | = 000040   | #4-62               |
| LF.TIM | = 000010   | #4-62               |
| LF.UNL | = 020000   | #4-62               |
| LF.X2P | = 000000   | #4-62               |
| LN.CLO | = 000000   | #4-62               |
| LN.DUM | = 000005   | #4-62               |
| LN.LOA | = 000004   | #4-62               |
| LN.LOO | = 000003   | #4-62               |
| LN.OAU | = 000003   | #4-62               |
| LN.OFF | = 000001   | #4-62               |
| LN.ON  | = 000000   | #4-62               |
| LN.OOP | = 000004   | #4-62               |
| LN.OPE | = 000001   | #4-62               |
| LN.REF | = 000002   | #4-62               |
| LN.SER | = 000002   | #4-62               |
| LN.STA | = 000017   | #4-62               |
| LN.SUB | = 000360   | #4-62               |
| LN.TRI | = 000006   | #4-62               |
| L.COST | 000015     | #4-62               |
| L.CTL  | 000012     | #4-62               |
| L.CVA  | 177776     | #4-62               |
| L.DDM  | 000002     | #4-62               |
| L.DDS  | 000004     | #4-62               |
| L.DLC  | 000003     | #4-62               |
| L.DLM  | 000006     | #4-62               |
| L.DLS  | 000010     | #4-62               |
| L.FLG  | 000000     | #4-62               |
| L.KRBA | 000016     | #4-62               |
| L.LEN  | = 000022   | #4-62               |
| L.MPF  | 000022     | #4-62               |
| L.NMST | 000020     | #4-62               |
| L.NSTA | 000014     | #4-62               |
| L.OWNR | 000021     | #4-62               |
| L.UNT  | 000013     | #4-62               |
| M\$PRO | = *****    | 5-86 6-168          |
| PS     | = ***** GX | *5-122              |



CELOG1 - COMM/EXEC EVFNT LOGGIN MACRO V05.03b Friday 28-Jun-85 18:19 <sup>F 13</sup>  
Table of contents

6- 56 LOG NETWORK EVENT

\$INTX7 - LEVEL 7 INTERRUPT EXIT

```

150 .SBTTL $INTX7 - LEVEL 7 INTERRUPT EXIT
151
152 ;+
153 **- $INTX7 - LEVEL 7 INTERRUPT EXIT
154
155 This routine is used by Driver ISRs that run at priority level
156 7 during some portion of Interrupt Service. Drivers ISRs may
157 only push one item on the stack during level 7 interrupt
158 service, and must call $INTSX (to drop priority and switch to
159 the system stack) to post any completions.
160
161 Inputs: (stack must be formatted as follows:)
162 R5 (Saved in line table)
163 * KINAR5 (Saved in line table for RSX-11M+ only)
164 KISAR5 (Saved in line table)
165 PS (Saved by interrupt)
166 PC (Saved by interrupt)
167 -
168
169 000042 $INTX7::
170 000042 012605 MOV (SP)+,R5 ;; Restore R5
171
172 .IF DF K$$$DAS ;; Different for RSX-11M+
173 ;; with Kernel Data Space enabled
174 MOV (SP)+,KINAR5 ;; Restore Kernel I-Space APR5
175 .ENDC
176
177 000044 012667 000000G MOV (SP)+,KISAR5 ;; Restore Kernel mapping register 5
178 000050 000002 RTI ;; Return from Interrupt

```

|     |     |                                                        |
|-----|-----|--------------------------------------------------------|
| 5-  | 68  | \$CMQIN - QUEUE A CHAIN OF CCBS TO A LIST              |
| 6-  | 101 | \$CMORM - REMOVE A CHAIN OF CCBS FROM A LIST           |
| 7-  | 142 | \$CNV18 - CONVERT TO 18-BIT UNIBUS ADDRESS             |
| 8-  | 205 | \$MVTBF - MOVE FROM MAPPED BUFFER TO UNMAPPED BUFFER   |
| 9-  | 256 | \$MVFBF - MOVE FROM UNMAPPED BUFFER TO A MAPPED BUFFER |
| 10- | 310 | \$CALLX - MAPPED SUBROUTINE CALL                       |
| 11- | 365 | \$CEACC - ACCESS BLOCK IN EXTENDED POOL                |
| 11- | 366 | \$CECAC - ACCESS BLOCK IN ALTERNATE EXTENDED POOL      |
| 12- | 415 | \$PDVID - PROCESS NAME TO PDV INDEX                    |
| 13- | 450 | \$CEMUL - UNSIGNED MULTIPLY                            |
| 14- | 481 | \$CEDIV - UNSIGNED DIVISION                            |

CESUB MACRO V05.03b Friday 28-Jun-85 18:20 Page 14  
 \$CEDIV - UNSIGNED DIVISION

```

481 .SBTTL $CEDIV - UNSIGNED DIVISION
482
483 :+ **- $CEDIV - UNSIGNED DIVISION
484 :
485 : THIS SUBROUTINE PERFORMS UNSIGNED DIVISION WHERE THE RESULTS ARE LESS THAN
486 : 16 BITS. IT IS INCLUDED HERE SO THAT WE CAN CONDITIONALISE THE CODE FOR
487 : EXTENDED INSTRUCTION SET.
488 :
489 : NOTE THAT IT IS THE CALLERS RESPONSIBILITY TO ENSURE THAT THE RESULTS WILL
490 : FIT IN 16 BITS.
491 :
492 : INPUTS:
493 : R0 = DIVIDEND
494 : R1 = DIVISOR
495 :
496 : OUTPUTS:
497 : R0 = QUOTIENT
498 : R1 = REMAINDER
499 : -
500
501 000576 $CEDIV::IF DF R$$EIS
502
503 MOV R1,-(SP) ; SAVE DIVISOR
504 MOV R0,R1 ; SET UP DIVIDEND
505 CLR R0 ; CLEAR HIGH ORDER WORD
506 DIV (SP)+,R0 ; PERFORM THE DIVISION
507 RETURN
508
509 .IFF ; DF R$$EIS
510
511 000576 CALLR $DIV ; CALL EXEC DIVIDE ROUTINE
512
513 .ENDC ; DF R$$EIS

```

```

76 .SBTTL $CCBGT - Allocate a Standard CCB
77
78 :+
79 :*- $CCBGT- Allocate a Standard CCB
80 :
81 : This subroutine is called to allocate a CCB from the CCB pool. If the
82 : assembly option is chosen, the CCB may be allocated from System
83 : Dynamic Memory if the CCB pool is empty.
84 :
85 : Inputs:
86 : None.
87 :
88 : Outputs:
89 : R4 contains the address of the allocated CCB
90 : C-Bit is CLEAR if the CCB was successful, allocated
91 : C-Bit is SET and R4=0 if the allocation failed
92 :-
93 000000 000241 $CCBGT::CLC ; Assume Success
94 000002 INHIB$;; Inhibit Interrupts (and save C-Bit)
95
96 .IF DF M$$PRO
97 CALL $MPLCK ;; Lock Access to CommExec Resources
98 .ENDC
99
100 000014 CALL CCBGT ;; Allocate a CCB
101
102 .IF DF M$$PRO
103 CALL a(SP)+ ;; Co-Routine return to unlock resources
104 .ENDC
105
106 000020 ENABL$;; Enable Interrupts (and load final C-Bit)
107 000024 RETURN

```

```

548 .SBTTL BUFGT - General Buffer Allocation Routine
549
550 *--BUFGT- General Buffer Allocation Routine
551
552 This subroutine is called within this module to allocate a
553 buffer from a doubly linked list. This routine is always executed
554 with interrupts inhibited (and Multi-Processor locks set).
555
556 Inputs:
557 R5 contains the pointer to listhead in Allocation Control Block
558
559 Outputs:
560 R2 contains the Virtual Address of buffer
561 R3 contains the APR Bias of buffer
562 C-Bit is CLEAR if the buffer was successfully allocated
563 C-Bit is SET if the allocation failed
564
565 Registers Modified:
566 R5
567 APR6 will be mapped to the buffer.
568
569 Notes:
570 Before this routine is called, the calling routine must have issued
571 a CLC instruction followed by an INHIB$. No other values must be
572 stored on the stack because if the allocation fails, the FAIL: code
573 will reach back on the stack and set the C-Bit in the saved PSW so
574 that when the calling routine executes an ENABL$ the C-Bit will be
575 set to indicate the allocation failure.
576
577
578 000610 000241 BUFGT: CLC ;; Assume success
579 000612 012503 MOV (R5)+,R3 ;; Get next buffer's APR Bias
580 000614 012502 MOV (R5)+,R2 ;; Get next buffer's Virtual Address
581 000616 001417 BEQ 10$;; If EQ, No more buffers available
582 000620 MAP R3 ;; Map APR6 to the new buffer
583 000624 016245 000002 MOV 2(R2),-(R5) ;; Unlink buffer from
584 000630 011243 MOV (R2),-(R5) ;; the free list
585
586 000632 010364 000014 MOV R3,C.BUF(R4) ;; Load the Buffer Bias
587 000636 010364 000024 MOV R3,C.BUF2(R4)
588 000642 010264 000016 MOV R2,C.BUF+2(R4) ;; Load the Buffer Virtual Address
589 000646 010264 000026 MOV R2,C.BUF2+2(R4)
590
591 000652 005345 DEC -(R5) ;; Decrement count of available buffers
592 000654 RETURN
593
594 000656 005225 10$: INC (R5)+ ;; Increment count of allocation failures
595
596 000660 005266 000002 FAIL: INC PRIOFF+2(SP) ;; Set C-Bit in saved PSW for ENABL$.
597 000664 000261 SEC ;; Indicate buffer allocation failure
598 000666 RETURN

```

```

76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93 000000 000241
94 000002
95
96
97
98
99
100 000014
101
102
103
104
105
106 000020
107 000024

.SBTTL $CCBGT - Allocate a Standard CCB
;+
; **-$CCBGT- Allocate a Standard CCB
;
; This subroutine is called to allocate a CCB from the CCB pool. If the
; assembly option is chosen, the CCB may be allocated from System
; Dynamic Memory if the CCB pool is empty.
;
; Inputs:
; None.
;
; Outputs:
; R4 contains the address of the allocated CCB
; C-Bit is CLEAR if the CCB was successfully allocated
; C-Bit is SET and R4=0 if the allocation failed
;-
$CCBGT::CLC ; Assume Success
 INHIB$;; Inhibit Interrupts (and save C-Bit)
 .IF DF M$$PRO
 CALL $MPLCK ;; Lock Access to CommExec Resources
 .ENDC
 CALL CCBGT ;; Allocate a CCB
 .IF DF M$$PRO
 CALL @($P)+ ;; Co-Routine return to unlock resources
 .ENDC
 ENAB$;; Enable Interrupts (and load final C-Bit)
 RETURN

```

```

548 .SBTTL BUFGT - General Buffer Allocation Routine
549
550 *
551 **--BUFGT- General Buffer Allocation Routine
552
553 This subroutine is called within this module to allocate a
554 buffer from a doubly linked list. This routine is always executed
555 with interrupts inhibited (and Multi-Processor locks set).
556
557 Inputs:
558 R5 contains the pointer to listhead in Allocation Control Block
559
560 Outputs:
561 R2 contains the Virtual Address of buffer
562 R3 contains the APR Bias of Buffer
563 C-Bit is CLEAR if the buffer was successfully allocated
564 C-Bit is SET if the allocation failed
565
566 Registers Modified.
567 R5
568 APR6 will be mapped to the buffer.
569
570 Notes:
571 Before this routine is called, the calling routine must have issued
572 a CLC instruction followed by an INHIB$. No other values must be
573 stored on the stack because if the allocation fails, the FAIL: code
574 will reach back on the stack and set the C-Bit in the saved PSW so
575 that when the calling routine executes an ENABL$ the C-Bit will be
576 set to indicate the allocation failure.
577
578 000766 000241
579 000770 012503
580 000772 012502
581 000774 001417
582 000776
583 001002 016245 000002
584 001006 011245
585
586 001010 010364 000014
587 001014 010364 000024
588 001020 010264 000016
589 001024 010264 000026
590
591 001030 005345
592 001032
593
594 001034 005225
595
596 001036 005266 000002
597 001042 000261
598 001044

```

```

BUFGT: CLC ;; Assume success
 MOV (R5)+,R3 ;; Get next buffer's APR Bias
 MOV (R5)+,R2 ;; Get next buffer's Virtual Address
 BEQ 10$;; If EQ, No more buffers available
 MAP R3 ;; Map APR6 to the new buffer
 MOV 2(R2),-(R5) ;; Unlink buffer from
 MOV (R2),-(R5) ;; the free list

 MOV R3,C.BUF(R4) ;; Load the Buffer Bias
 MOV R3,C.BUF2(R4)
 MOV R2,C.BUF+2(R4) ;; Load the Buffer Virtual Address
 MOV R2,C.BUF2+2(R4)

 DEC -(R5) ;; Decrement count of available buffers
 RETURN

10$: INC (R5)+ ;; Increment count of allocation failures

FAIL: INC PRI0FF+2(SP) ;; Set C-Bit in saved PSW for ENABL$.
 SEC ;; Indicate buffer allocation failure
 RETURN

```



```

71
72 ; LOCAL MACRO DEFINITIONS
73 ;
74
75 .IF DF K$$DAS
76
77 .MACRO DDFDF,FNC,ARG1
78 MOV R3,-(SP) ; SAVE R3
79 MOV #FNC,R3 ; GET FUNCTION CODE
80 .IF B <ARG1>
81 BR DDCM1 ; COMMON PROCESSING FOR NON-CCB CONTROL FUNCTION
82 .IFF
83 .IF IDN <ARG1>,<MDC>
84 BR DDCM2 ; COMMON PROCESSING FOR MODEM CONTROL FUNCTIONS
85 .IFF
86 .IF IDN <ARG1>,<SUB>
87 BR DDCM4
88 .IFF
89 BR DDCM3 ; COMMON PROCESSING FOR CONTROL FUNCTIONS WITH CCB
90 .ENDC
91 .ENDC
92 .ENDC
93 .ENDM DDFDF
94
95 .IFF ; DF K$$DAS
96
97 .MACRO DDFDF,FNC,ARG1
98 .IF B <ARG1>
99 JSR R3,DDCM1 ; COMMON PROCESSING FOR NON-CCB CONTROL FUNCTION
100 .IFF
101 .IF IDN <ARG1>,<MDC>
102 JSR R3,DDCM2 ; COMMON PROCESSING FOR MODEM CO. ROL FUNCTIONS
103 .IFF
104 .IF IDN <ARG1>,<SUB>
105 JSR R3,DDCM4
106 .IFF
107 JSR R3,DDCM3 ; COMMON PROCESSING FOR CONTROL FUNCTIONS WITH CCB
108 .ENDC
109 .ENDC
110 .ENDC
111 10$: FNC
112 .ENDM
113
114 .ENDC ; DF K$$DAS

```

|                |                 |                 |                 |                |
|----------------|-----------------|-----------------|-----------------|----------------|
| A\$CHK= 000000 | CS.ENB= 000020  | D\$SYNM= 000000 | I\$SRDN= 000000 | M\$FCS= 000000 |
| A\$CPS= 000000 | CS.ERR= 100000  | E\$XPR= 000000  | K\$CNT= 177546  | M\$MGE= 000000 |
| A\$SPR= 000000 | CS.FTL= 001000  | FC.CCP= 000020  | K\$CSR= 177546  | M\$NET= 000000 |
| A\$TRP= 000000 | CS.HCR= 000001  | FC.CTL= 000006  | K\$LDL= 000000  | M\$DVR= 000000 |
| CB.CCB= 000002 | CS.HFE= 002000  | FC.KCP= 000016  | K\$TPS= 000074  | N\$ACC= 000001 |
| CB.DDM= 000040 | CS.LST= 040000  | FC.KIL= 000004  | LDLP = 000000   | N\$BUF= 000001 |
| CB.DLC= 000020 | CS.MTL= 004000  | FC.MAN= 000024  | LF.ACT= 100000  | N\$LDV= 000001 |
| CB.RDB= 000004 | CS.RNG= 000010  | FC.MLD= 000026  | LF.BRD= 000400  | N\$MCP= 000001 |
| CB.SDB= 000010 | CS.RDV= 000004  | FC.PCT= 000030  | LF.BWT= 000007  | N\$MLL= 000001 |
| CB.SLI= 000100 | CS.RSN= 010000  | FC.PWR= 000022  | LF.ENA= 002000  | N\$MOV= 000010 |
| CB.XLB= 000001 | CS.SHU= 000001  | FC.RCE= 000002  | LF.LPB= 001000  | N\$NCT= 000001 |
| CC.LLC= 000200 | CS.SID= 000902  | FC.RCP= 000014  | LF.MDC= 000100  | N\$PEM= 000001 |
| CE.ABO= 100362 | CS.STR= 000004  | FC.TIM= 000010  | LF.MFL= 004000  | PR7 = ***** GX |
| CE.DAO= 100346 | CS.SUC= 000001  | FC.XCP= 000012  | LF.MTP= 000020  | PS = ***** GX  |
| CE.DIS= 100366 | CS.TMD= 020000  | FC.XME= 000000  | LF.PAC= 000200  | P\$P45= 000000 |
| CE.ERR= 100370 | CS.XUR= 000004  | FS.AST= 000000  | LF.RDY= 040000  | P\$WRD= 000000 |
| CE.ILN= 100350 | C\$CKP= 000000  | FS.CIB= 002000  | LF.REA= 010000  | Q\$OPT= 000010 |
| CE.LTD= 100356 | C\$ORE= 000400  | FS.CRA= 001000  | LF.SER= 000040  | R\$BDR= 000000 |
| CE.MOP= 100372 | C\$RSH= 177564  | FS.DIS= 013000  | LF.TIM= 000010  | R\$K11= 000001 |
| CE.NTE= 100361 | C.ADD= 000034   | FS.DVC= 001000  | LF.UNL= 020000  | R\$SND= 000000 |
| CE.RTE= 100376 | C.BID= 000003   | FS.ENB= 012000  | LF.X2P= 000000  | R\$11M= 000000 |
| CE.SRC= 100364 | C.BUF= 000014   | FS.EXI= 001000  | LN.CLO= 000000  | SF.ACT= 000200 |
| CE.STP= 100352 | C.BUF1= 000014  | FS.GET= 006000  | LN.DUM= 000005  | SF.ENA= 000100 |
| CE.TMC= 100354 | C.BUF2= 000024  | FS.HLT= 000000  | LN.LDA= 000004  | SF.LPB= 000004 |
| CE.TMD= 100374 | C.CNT= 000020   | FS.INI= 000000  | LN.LDD= 000003  | SF.MFL= 000040 |
| CE.UNS= 100344 | C.CNT1= 000020  | FS.KIL= 000000  | LN.OAU= 000003  | SF.PAC= 000020 |
| CF.CHN= 000001 | C.CNT2= 000030  | FS.LCL= 100000  | LN.OFF= 000001  | SF.REA= 000010 |
| CF.EOM= 000004 | C.FLG= 000022   | FS.LTM= 001000  | LN.DN = 000000  | SF.SER= 000001 |
| CF.HDR= 000020 | C.FLG1= 000022  | FS.MNT= 004000  | LN.DOP= 000004  | SF.SVC= 000002 |
| CF.LB = 100000 | C.FLG2= 000032  | FS.MSN= 014000  | LN.OPE= 000001  | SF.UNL= 000040 |
| CF.LIN= 000002 | C.FNC= 000010   | FS.REA= 001000  | LN.REF= 000002  | S\$WRG= 000000 |
| CF.SOM= 000010 | C.LIN= 000006   | FS.RET= 000000  | LN.SER= 000002  | S\$YSZ= 007600 |
| CF.SYN= 000040 | C.LNK= 000000   | FS.REZ= 003000  | LN.STA= 000017  | S.CDST= 000001 |
| CF.TRN= 000100 | C.MOD= 000011   | FS.RLB= 002000  | LN.SUB= 000360  | S.FLG= 000000  |
| CM.CIR= 000002 | C.NSP= 000004   | FS.RNG= 011000  | LN.TRI= 000006  | S.LEN= 000004  |
| CM.FMT= 100000 | C.PROD= 000042  | FS.RST= 000000  | L\$ASG= 000000  | S.NMST= 000002 |
| CM.HRD= 000002 | C.RSV= 000002   | FS.RTN= 001000  | L\$DRV= 000000  | S.DWNR= 000003 |
| CM.LIN= 000000 | C.STA= 000007   | FS.SET= 005000  | L\$P11= 000001  | T\$KMG= 000000 |
| CM.LOD= 000001 | C.STS= 000012   | FS.SFC= 005000  | L\$11R= 000000  | T\$MIN= 000000 |
| CM.XLO= 000004 | C.URM= 177776   | FS.SFR= 006000  | L.CDST= 000015  | V\$CTR= 001000 |
| CP.DCF= 000040 | C.XACP= 000004  | FS.SFS= 004000  | L.CTL= 000012   | X\$DBT= 000000 |
| CP.HDL= 000007 | C.XID= 000035   | FS.SPW= 040000  | L.CVA= 177776   | ZF.CDU= 001000 |
| CP.PS = 177400 | C.XLEN= 000044  | FS.STM= 000000  | L.DDM= 000002   | ZF.DDM= 000001 |
| CP.PSI= 000200 | C.XPLI= 000040  | FS.STP= 002000  | L.DDS= 000004   | ZF.DIA= 004000 |
| CP.XCF= 000100 | C.XPT= 000034   | FS.STR= 001000  | L.DLC= 000903   | ZF.DLC= 000002 |
| CP.2FR= 000030 | C.XSVC= 000042  | FS.TRM= 003000  | L.DLM= 000006   | ZF.DVP= 100000 |
| CS.ABO= 000100 | C.XTC= 000037   | FS.WLB= 001000  | L.DLS= 000010   | ZF.INI= 040000 |
| CS.BRO= 000002 | C.X25= 000036   | FS.XKL= 002000  | L.FLG= 000000   | ZF.KMX= 000020 |
| CS.BUF= 000200 | DDCM1= 000140R  | FS.XDF= 010000  | L.KRBA= 000016  | ZF.LLC= 000004 |
| CS.CES= 000002 | DDCM2= 000150R  | FS.XDN= 007000  | L.LEN= 000022   | ZF.LMC= 000100 |
| CS.CHN= 000010 | DDCM3= 000150R  | FS.ZER= 002000  | L.MPF= 000022   | ZF.MAN= 020000 |
| CS.CMP= 000200 | DDCM4= 000156R  | F\$SLVL= 000001 | L.NMST= 000020  | ZF.MFL= 000010 |
| CS.DCR= 000400 | DDMDSP= 000206R | G\$TPP= 000000  | L.NSTA= 000014  | ZF.MTM= 000400 |
| CS.DEF= 000004 | D\$BUG= 177514  | G\$TSS= 000000  | L.DWNR= 000021  | ZF.MUX= 000040 |
| CS.DEV= 000002 | D\$ISK= 000000  | G\$TTK= 000000  | L.UNT= 000013   | ZF.PSE= 002000 |
| CS.DIS= 000040 | D\$L1= 000001   | G\$WRD= 000000  | M\$CRB= 000124  | ZF.SLI= 010000 |
| CS.ENA= 000001 | D\$YNC= 000000  | J\$RAR= 000000  | M\$CRX= 000000  | ZF.TIM= 000200 |

|            |                |                   |                             |
|------------|----------------|-------------------|-----------------------------|
| 174 000000 | \$DDXME::DDFDF | FC.XME,SUB        | : TRANSMIT ENABLE           |
| 175 000006 | \$DDRCE::DDFDF | FC.RCE,SUB        | : RECEIVE ENABLE            |
| 176        |                |                   |                             |
| 177 000014 | \$DDMAN::DDFDF | FC.MAN,SUB        | : NETWORK MANAGEMENT        |
| 178        |                |                   |                             |
| 179 000022 | \$DDXKL::DDFDF | FC.KIL+FS.XKL,CCB | : TRANSMIT KILL             |
| 180 000030 | \$DDCRA::DDFDF | FC.KIL+FS.CRA,CCB | : RECEIVE KILL              |
| 181 000036 | \$DDKIL::DDFDF | FC.KIL+FS.KIL,CCB | : RECEIVE AND TRANSMIT KILL |
| 182 000044 | \$DDSTR::DDFDF | FC.CTL+FS.STR,CCB | : START                     |
| 183 000052 | \$DDSTP::DDFDF | FC.CTL+FS.STP,CCB | : STOP                      |
| 184 000060 | \$DDGET::DDFDF | FC.CTL+FS.GET,CCB | : GET CHARACTERISTICS       |
| 185 000066 | \$DDSET::DDFDF | FC.CTL+FS.SET,CCB | : SET CHARACTERISTICS       |
| 186        |                |                   |                             |
| 187 000074 | \$DDRNG::DDFDF | FC.CTL+FS.RNG,MDC | : LOOK FOR RING             |
| 188 000102 | \$DDENB::DDFDF | FC.CTL+FS.ENB,MDC | : ENABLE LINE               |
| 189 000110 | \$DDDIS::DDFDF | FC.CTL+FS.DIS,MDC | : DISABLE LINE              |
| 190        |                |                   |                             |
| 191 000116 | \$DDXON::DDFDF | FC.CTL+FS.XON     | : XON                       |
| 192 000124 | \$DDXOF::DDFDF | FC.CTL+FS.XOF     | : XOFF                      |
| 193 000132 | \$DDMSN::DDFDF | FC.CTL+FS.MSN     | : SENSE MODEM STATUS        |
| 194        |                |                   |                             |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE    | REFERENCES |
|---------|----------|------------|
| C.FNC   | 000010   | *10-281    |
| C.LIN   | 000006   | *11-350    |
| C.STS   | 000012   | *10-290    |
| DDCM1   | 000140 R | *10-307    |
| DDCM2   | 000150 R | *11-361    |
| DDCM3   | 000254 R | *13-456    |
| DDCM4   | 000262 R | *14-497    |
| DDMDS   | 000312 R | *14-500    |
| FC.CCP  | = 000020 | *14-503    |
| FC.CTL  | = 000006 | *14-510    |
| FC.KCP  | = 000016 | *11-342    |
| FC.KIL  | = 000004 |            |
| FC.MAN  | = 000024 |            |
| FC.RCE  | = 000002 |            |
| FC.RCP  | = 000014 |            |
| FC.XCP  | = 000012 |            |
| FC.XME  | = 000000 |            |
| FS.AST  | = 000000 |            |
| FS.CRA  | = 001000 |            |
| FS.DIS  | = 013000 |            |
| FS.ENB  | = 012000 |            |
| FS.GET  | = 006000 |            |
| FS.KIL  | = 000000 |            |
| FS.MSN  | = 014000 |            |
| FS.RNG  | = 011000 |            |
| FS.SET  | = 005000 |            |
| FS.STP  | = 002000 |            |
| FS.STR  | = 001000 |            |
| FS.XKL  | = 002000 |            |
| FS.XOF  | = 010000 |            |
| FS.XON  | = 007000 |            |
| IS\$AS  | = *****  |            |
| K\$SDAS | = *****  |            |
| LF.ACT  | = 100000 |            |
| LF.BRO  | = 000400 |            |
| LF.BWT  | = 000007 |            |
| LF.ENA  | = 002000 |            |
| LF.LPB  | = 001000 |            |
| LF.MDC  | = 000100 |            |
| LF.MFL  | = 004000 |            |
| LF.MTP  | = 000020 |            |
| LF.PAC  | = 000200 |            |
| LF.PDY  | = 040000 |            |
| LF.REA  | = 010000 |            |
| LF.SER  | = 000040 |            |
| LF.TIM  | = 000010 |            |
| LF.UNL  | = 020000 |            |
| LF.X2P  | = 000000 |            |
| LN.CLO  | = 000000 |            |
| LN.DUM  | = 000005 |            |
| LN.LOA  | = 000004 |            |

```

236 .SBTTL DDCM2 - COMMON PROCESS FOR MODEM CONTROL FUNCTIONS
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292

```

```

 .SBTTL DDCM2 - COMMON PROCESS FOR MODEM CONTROL FUNCTIONS
 **--DDCM2 - COMMON PROCESS FOR MODEM CONTROL FUNCTIONS
 IF THE SPECIFIED LINE HAS MODEM CONTROL ENABLED THE
 FUNCTION REQUEST IS DERAILED TO THE MODEM CONTROLLER.
 ON RETURN FROM THE MODEM CONTROLLER THE REQUEST IS
 ROUTED TO THE PROPER DDM AS USUAL. IF THE LINE IS
 HARDWIRED THE REQUEST IS ROUTED DIRECTLY TO THE DDM.
 INPUTS:
 R3 = ADDRESS OF FUNCTION CODE
 R4 = ADDRESS OF CCB
 STACK CONTAINS:
 O0(SP) = CALLER'S R3
 O2(SP) = RETURN ADDRESS TO CALLING DLC
 OUTPUTS TO MODEM CONTROLLER:
 R3 = SUBFUNCTION CODE
 R5 = LINE TABLE ADDRESS
 OUTPUTS TO DDM:
 R3 = SUBFUNCTION CODE
 R4 = SAME AS ON ENTRY
 R5 = LINE TABLE ADDRESS
 OUTPUTS ON RETURN TO CALLER:
 C-BIT CLEAR - FUNCTION PERFORMED SYNCHRONOUSLY
 C-BIT SET - WAIT FOR ASYNCHRONOUS REPORT OF FUNCTION COMPLETION
 REGISTERS ACROSS CALL:
 R2,R3,R5 - PRESERVED
 R4 - MODIFIED
 -
DDCM2:
 .IF DF X$$MDC
 .IF DF K$$DAS
 MOV R3,$DDFNC ; STORE THE FUNCTION CODE IN DATA AREA
 MOV #DDFNC,R3 ; POINT R3 AT FUNCTION CODE
 .ENDC
 MOV (R3),C.FNC(R4) ; PUT FUNCTION CODE IN CCB
 MOV R2,-(SP) ; SAVE R2
 .IF DF N$$1LN
 MOV $SL7MA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE
 .IFF
 MOVB C.LIN(R4),R2 ; GET SYSTEM LINE NUMBER
 ASL R2 ; FORM WORD INDEX
 ADD $SL7MA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE

```

CEDDMN      CREATED BY    MACRO    ON 28-JUN-85 AT 18:18

PAGE 3      6 10

MACRO CROSS REFERENCE

CREF    04.00

| MACRO NAME | REFERENCES |
|------------|------------|
|------------|------------|

|         |        |      |
|---------|--------|------|
| CALLR   | #5-65  |      |
| CCBDF\$ | #5-64  | 5-66 |
| DDFDF   | #6-97  |      |
| ENABL\$ | #5-63  |      |
| INHIB\$ | #5-63  |      |
| PDVDF\$ | #5-64  | 5-67 |
| RESRG   | #5-63  |      |
| RETURN  | 15-578 |      |
| SAJRG   | #5-63  |      |
| SLTDF\$ | #5-64  | 5-68 |

CE DLC      CREATED BY MACRO ON 28-JUN-85 AT 18:19      PAGE 3      G 11

SYMBOL CROSS REFERENCE      CREF      04.00

SYMBOL      VALUE      REFERENCES

|         |            |        |
|---------|------------|--------|
| Z.SCH   | 000007     | #4-63  |
| \$ASCMP | 000034 RG  | #6-140 |
| \$CCBGT | = ***** GX | 6-140  |
| \$CTCMP | 000066 RG  | #7-178 |
| \$KLCMP | 000076 RG  | #7-181 |
| \$LLCTA | = ***** GX | 7-192  |
| \$PDQUE | = ***** GX | 7-208  |
| \$RCCMP | 000106 RG  | #7-184 |
| \$SLTMA | = ***** GX | 5-103  |
| \$STDLC | 000000 RG  | #5-90  |
| \$STDLC | 000004 RG  | #5-96  |
| \$XMCMP | 000056 RG  | #7-175 |

CELLC CREATED BY MACRO ON 28-JUN-85 AT 18:19 PAGE 2 G 12  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL   | VALUE       | REFERENCES  |
|----------|-------------|-------------|
| R\$\$11D | = *****     | 4-61        |
| R\$\$11M | = 000000    | 4-61        |
| R\$\$11S | = *****     | 4-61        |
| SF.ACT   | = 000200    | #4-62       |
| SF.ENA   | = 000100    | #4-62       |
| SF.LPB   | = 000004    | #4-62       |
| SF.MFL   | = 000040    | #4-62       |
| SF.PAC   | = 000020    | #4-62       |
| SF.REA   | = 000010    | #4-62       |
| SF.SER   | = 000001    | #4-62       |
| SF.SVC   | = 000002    | #4-62       |
| SF.UNL   | = 000040    | #4-62       |
| S.COST   | = 000001    | #4-62       |
| S.FLG    | = 000000    | #4-62       |
| S.LEN    | = 000004    | #4-62       |
| S.NMST   | = 000002    | #4-62       |
| S.OWNR   | = 000003    | #4-62       |
| X\$MBCB  | = *****     | 4-61 4-61   |
| ZF.COY   | = 001000    | #4-61       |
| ZF.DDM   | = 000001    | #4-61       |
| ZF.DIA   | = 004000    | #4-61       |
| ZF.DLC   | = 000.02    | #4-61       |
| ZF.DVP   | = 100000    | #4-61       |
| ZF.INI   | = 040000    | #4-61       |
| ZF.KMX   | = 000020    | #4-61       |
| ZF.LLC   | = 000004    | #4-61       |
| ZF.LMC   | = 000100    | #4-61       |
| ZF.MAN   | = 020000    | #4-61       |
| ZF.MFL   | = 000010    | #4-61       |
| ZF.MTM   | = 000400    | #4-61       |
| ZF.MUX   | = 000040    | #4-61       |
| ZF.PSE   | = 002000    | #4-61       |
| ZF.SLI   | = 010000    | #4-61       |
| ZF.TIM   | = 000200    | #4-61       |
| ZF.X3P   | = 000000    | #4-61       |
| ZS.ASN   | = 100000    | #4-61       |
| ZS.BSY   | = 140000    | #4-61       |
| Z.AVL    | = 000014    | #4-61       |
| Z.DAT    | = 000016    | 7-225 #4-61 |
| Z.DSP    | = 000000    | 4-61        |
| Z.FLG    | = 000010    | #4-61       |
| Z.LEN    | = 000016    | #4-61       |
| Z.LLN    | = 000006    | #4-61       |
| Z.MAP    | = 000020    | 5-117 #4-61 |
| Z.NAM    | = 000004    | #4-61       |
| Z.PCB    | = 000012    | #4-61       |
| Z.SCH    | = 000007    | #4-61       |
| \$CMPDV  | = ***** GX  | 5-111 7-219 |
| \$DL CRS | = 000074 RG | #6-167      |
| \$LL CAL | = 000146 RG | #7-221      |
| \$LL CLC | = 000126 RG | #7-216      |
| \$LL CRG | = 000000 RG | #5-85       |



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54

000000

```
.IIF NDF N$SEVL .TITLE CELOG - COMM/EXEC EVENT LOGGING ROUTINES
.IIF DF N$SEVL .TITLE CELOG1 - COMM/EXEC EVENT LOGGING ROUTINES
.IDENT /V05.00/
```

```

: COPYRIGHT (C) 1980, 1982, 1983, 1985 BY
: DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
```

```

: THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
: SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
: INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR
: ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE
: MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH
: SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE
: TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN
: IN DEC.
```

```

: THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
: NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
: EQUIPMENT CORPORATION.
```

```

: DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
: ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
```

```

: MODULE DESCRIPTION
```

```

: CEX EVENT LOGGING ROUTINES
```

```

: DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
```

```

: IDENT HISTORY:
```

```

: 1.00 30-JAN-80
:
: 3.00 16-APR-82
: DECNET-11M V3.1
: DECNET-11M-PLUS V1.1
:
: 4.00 07-NOV-83
: DECNET-11M V4.0
: DECNET-11M-PLUS V2.0
:
: 5.00 22-JUL-85
: DECnet-11M/S V4.2
: DECnet-11M-Plus V3.0
: DECnet-Micro/RSX V1.0
```

```
.MCALL PDVDF$,SAVRG,RESRG
```

```
PDVDF$; DEFINE PDV OFFSETS
```

```

180 .SBTTL $PDSPL - SET UP A PROCESS LEVEL AND DISPATCH TO IT
181
182 ;+
183 ;**-$PDSPL-SET UP A PROCESS LEVEL AND DISPATCH TO IT
184 ;
185 ; This subroutine is called to set up the level of a process
186 ; and dispatch to it. This routine assumes that the stack is formatted
187 ; properly for a pass through the scheduler when the dispatched process
188 ; exits.
189 ;
190 ; Inputs:
191 ;
192 ; R2 = PDV Index (word index)
193 ; R3 = Address of function code
194 ; R4 = Address of a CCB (optional)
195 ; R5 = Address of a Line Table (optional)
196 ;
197 ; Outputs:
198 ; The new process execution priority is established and dispatch
199 ; is performed by the routine '$PDDSP'.
200 ;
201 ;-
202
203 000052 $PDSPL::MTPS #PR7 ::: Disable Interrupts
204 000060 010201 MOV R2,R1 ::: Copy PDV Index
205 000062 066701 ADD $PDVTA,R1 ::: Point into PDV Index Table
206 000066 011101 MOV (R1),R1 ::: Get address of process' PDV
207
208 .IF DF L$$$I1
209 MFPS -(SP) ::: Get current priority level
210 BICB Z.SCH(R1),(SP) ::: Set priority of process
211 MTPS (SP)+ :: ...
212
213 .IFF
214 000070 146167 000007 000000G BICB Z.SCH(R1),PS ;; Set priority of process
215 .ENDC
216 :: DF L$$$I1

```

1 .IIF NDF R\$\$EIS .TITLE CESUB  
2 .IIF OF R\$\$EIS .TITLE CESUB1  
3 .IDENT /V05.00/  
4  
5  
6  
7  
8  
9

10 COPYRIGHT (C) 1978,1979,1980,1981, 1982, 1983, 1985 BY  
11 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.  
12  
13  
14  
15

16 THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
17 ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
18 INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
19 COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
20 OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
21 TRANSFERRED.  
22

23 THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
24 AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
25 CORPORATION.  
26

27 DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
28 SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
29

30 MODULE DESCRIPTION  
31

32 CEX SUBROUTINES  
33  
34  
35  
36  
37  
38  
39

40 DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING  
41

42 IDENT HISTORY:  
43

- 44 1.00 10-FEB-78  
45 VERSION 2.0 RELEASE  
46  
47 2.00 14-DEC-79  
48 DECNET-11M/S V3.0  
49 DECNET-11M-PLUS V1.0  
50  
51 3.00 16-APR-82  
52 DECNET-11M V3.1  
53 DECNET-11M-PLUS V1.1  
54  
55 4.00 07-NOV-83  
56 DECNET-11M V4.0  
57 DECNET-11M-PLUS V2.0  
58  
59 5.00 22-JUL-85  
60 DECnet-11M/S V4.2  
61 DECnet-11M-Plus V3.0  
62 DECnet-Micro/RSX V1.0  
63  
64  
65  
66  
67

```

515 .IF DF M$$PRO
516
517 .SBTTL $MPLCK - MULTI-PROCESSOR LOCK ROUTINE
518
519 ;+
520 **-$MPLCK-MULTI-PROCESSOR LOCK ROUTINE
521
522 THIS ROUTINE WILL PERFORM A SPIN LOCK ON THE COMMEXEC LOCK ($CRESL)
523 AND THEN CO-CALL THE CALLER. ON RETURN IT WILL UNLOCK THE COMMEXEC
524 LOCK AND RETURN. NOTE THAT WE PERFORM THE LOCK OPERATION OURSELVES
525 SO THAT WE CAN CORRECTLY MANIPULATE THE CACHE.
526
527 -$MPLCK::ASRB $CRESL ; SPIN ON THE LOCK
528 BCC $MPLCK ; UNTIL WE HAVE ACCESS
529 CACHE$ FLUSH ; FLUSH THE CACHE OF STALE DATA
530
531 CALL @($P)+ ; CO-CALL THE CALLER BACK
532
533 MOVB #1,$CRESL ; RESET THE LOCK
534
535 RETURN

```

```

109 .SBTTL $LDBGT - Get a Large Data Buffer
110
111 :+
112 :*- $LDBGT- Get a Large Data Buffer
113 :
114 : This subroutine is called to allocate a fixed length Large Data
115 : Buffer. Large Data Buffers come out of the same pool as Receive
116 : Data Buffers. The difference is that Large Data Buffers are for
117 : use by non-DDM processes and there must be a minimum number of
118 : buffers remaining in the pool or the allocation will fail. The
119 : minimum number reserved for DDM processes is the RDB threshold.
120 :
121 : Inputs:
122 : None.
123 :
124 : Outputs:
125 : R4 contains the address of buffers allocated CCB
126 : C-Bit is CLEAR if the CCB/LDB was successfully allocated
127 : C-Bit is SET and R4=0 if the allocation failed
128 :
129 : Note:
130 : The caller's APR6 is preserved across this call.
131 :
132 000026 026767 000000G 000000G $LDBGT::CMP $RDBCT,$RDBTH ; Is pool minimum threshold exceeded?
133 000034 103003 $RDBGT ; If HIS, No ... go allocate buffer
134 000036 0C267 000000G INC $LDBAF ; Increment count of allocation failures
135 000042 RETURN ; Return with C-Bit SET

```

```

600 .SBTTL BUFRT - General Buffer Deallocation Routine
601
602 ;+
603 ;**~BUFRT- General Buffer Deallocation Routine
604 ;
605 ; This routine is called within this module to deallocate a buffer
606 ; to a doubly linked list. This routine is always called with
607 ; interrupts inhibited (and Multi-Processor locks set).
608
609 ; Inputs:
610 ; R2 contains the Virtual Address of buffer
611 ; R3 contains the APR Bias of buffer
612 ; R5 points to the buffer count in allocation control block
613
614 ; Outputs:
615 ; None.
616 ;+
617 000670 005225 BUFRT: INC (R5)+ ;:: Increment count of available buffers
618 000672 SAVMAP ;:: Save current APR6 mapping
619 000676 MAP R3 ;:: Map APR6 to the buffer being returned
620 000702 012512 MOV (R5)+,(R2) ;:: Link buffer to start
621 000704 012562 000002 MOV (R5)+,2(R2) ;:: of the free list
622 000710 RESMAP ;:: Restore APR6 mapping
623 000714 010245 MOV R2,-(R5) ;:: Set new first buffer
624 000716 010345 MOV R3,-(R5) ;:: pointer
625 000720 RETURN

```

```

109 .SBTTL $LDBGT - Get a Large Data Buffer
110 ;+
111 ;**-$LDBGT- Get a Large Data Buffer
112 ;
113 ; This subroutine is called to allocate a fixed length Large Data
114 ; Buffer. Large Data Buffers come out of the same pool as Receive
115 ; Data Buffers. The difference is that Large Data Buffers are for
116 ; use by non-DDM processes and there must be a minimum number of
117 ; buffers remaining in the pool or the allocation will fail. The
118 ; minimum number reserved for DDM processes is the RDB threshold.
119 ;
120 ; Inputs:
121 ; None.
122 ;
123 ; Outputs:
124 ; R4 contains the address of buffers allocated CCB
125 ; C-Bit is CLEAR if the CCB/LDB was successfully allocated
126 ; C-Bit is SET and R4=0 if the allocation failed
127 ;
128 ; Note:
129 ; The caller's APR6 is preserved across this call.
130 ; -
131
132 000026 026767 000000G 000000G $LDBGT::CMP $RDBCT,$RDBTH ; Is pool minimum threshold exceeded?
133 000034 103003 BHIS $RDBGT ; If HIS, No ... go allocate buffer
134 000036 005267 000000G INC $LDBAF ; Increment count of allocation failures
135 000042 RETURN ; Return with C-Bit SET

```

```

600 .SBTTL BUFRT ~ General Buffer Deallocation Routine
601 +
602 **--BUFRT- General Buffer Deallocation Routine
603
604 This routine is called within this module to deallocate a buffer
605 to a doubly linked list. This routine is always called with
606 interrupts inhibited (and Multi-Processor locks set).
607
608 Inputs:
609 R2 contains the Virtual Address of buffer
610 R3 contains the APR Bias of buffer
611 R5 points to the buffer count in allocation control block
612
613 Outputs:
614 None.
615 +
616
617 001046 005225 BUFRT: INC (R5)+ ;;; Increment count of available buffers
618 001050 SAVMAP ;;; Save current APR6 mapping
619 001054 MAP R3 ;;; Map APR6 to the buffer being returned
620 001060 012512 MOV (R5)+,(R2) ;;; Link buffer to start
621 001062 012562 000002 MOV (R5)+,2(R2) ;;; of the free list
622 001066 RESMAP ;;; Restore APR6 mapping
623 001072 010245 MOV R2,-(R5) ;;; Set new first buffer
624 001074 010345 MOV R3,-(R5) ;;; pointer
625 001076 RETURN

```



```

116 .IF NDF X$$NDM
117 .SBTTL $DD??? - DLC TO DDM REQUESTS
118
119 +
120 GENERAL REQUESTS WITH CCB
121
122 FORMAT OF CALL:
123 CALL $DD???
124
125 INPUTS:
126 R4 = ADDRESS OF FIRST CCB IN CHAIN
127 (ALL CCBS MUST CONTAIN A VALID SLN)
128
129 OUTPUTS TO DDM:
130 R4 = ADDRESS OF FIRST CCB IN CHAIN
131 R5 = ADDRESS OF DDM LINE TABLE
132 R2 & R3 - AVAILABLE FOR USE WITHOUT SAVING
133
134 REGISTERS ACROSS CALL:
135 R0,R1 - MUST BE PRESERVED BY DDM IF USED
136 R2,R3,R5 - PRESERVED BY COMM EXEC
137 R4 - MAY BE MODIFIED
138
139 ON RETURN TO DLC:
140 C-BIT CLEAR - REQUEST HAS COMPLETED SYNCHRONOUSLY
141 R4 = ADDRESS OF CCB
142
143 C-BIT SET - REQUEST WILL COMPLETE ASYNCHRONOUSLY
144 R4 = 0
145
146 -
147
148 +
149 SPECIAL REQUESTS WITHOUT CCB
150
151 FORMAT OF CALL:
152 CALL $DD???
153
154 INPUTS:
155 R3 = SYSTEM LINE NUMBER
156 R4 = OPTIONAL CALLING PARAMETER TO DDM
157
158 OUTPUTS TO DDM:
159 R4 = OPTIONAL CALLING PARAMETER FROM DLC
160 R5 = ADDRESS OF DDM LINE TABLE
161 R2 & R3 - AVAILABLE FOR USE WITHOUT SAVING
162
163 REGISTERS ACROSS CALL:
164 R0,R1 - MUST BE PRESERVED BY DDM IF USED
165 R2,R3,R5 - PRESERVED BY COMM EXEC
166 R4 - MAY BE MODIFIED
167
168 ON RETURN TO DLC:
169 C-BIT ALWAYS CLEAR
170 R4 = OPTIONAL RETURNING PARAMETER FROM DDM
171
172 -

```

|                |                   |                  |                  |                    |
|----------------|-------------------|------------------|------------------|--------------------|
| ZF.X3P= 000000 | Z.MAP 000020      | \$DDDIS 000110RG | \$DDRNG 000074RG | \$DDXON 000116RG   |
| ZS.ASN= 100000 | Z.NAM 000004      | \$DDENB 000102RG | \$DDSET 000066RG | \$LLCSP 000374RG   |
| ZS.BSY= 140000 | Z.PCB 000012      | \$DDGET 000060RG | \$DDSPC 000374RG | \$PDDSP= ***** GX  |
| Z.AVL 000014   | Z.SCH 000007      | \$DDKCP 000344RG | \$DDSTP 000052RG | \$PDQU1= ***** GX  |
| Z.DAT 000016   | \$CCBGT= ***** GX | \$DDKIL 000036RG | \$DDSTR 000044RG | \$PDVTA= ***** GX  |
| Z.DSP 000000   | \$CCBRT= ***** GX | \$DDMAN 000014RG | \$DDXKL 000022RG | \$SLTMA= ***** GX  |
| Z.FLG 000010   | \$DDAST 000274RG  | \$DDMSN 000132RG | \$DDXME 000000RG | \$STDDM 000404RG   |
| Z.LEN = 000016 | \$DDCCP 000354RG  | \$DDRCE 000006RG | \$DDXMP 000324RG | \$STDD1 000410RG   |
| Z.LLN 000006   | \$DDCRA 000030RG  | \$DDRCP 000334RG | \$DDXOF 000124RG | .\$\$\$\$.= 000034 |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000432 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 9  
Work file writes: 12  
Size of work file: 17461 Words ( 69 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:18.54  
SY:CEDDM.V2,[130,134]CEDDM/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]CEDDM

196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234

.SBTTL DDCM1 - COMMON PROCESS FOR SPECIAL NON-CCB FUNCTIONS

\*  
\*\*-DDCM1 - COMMON PROCESS FOR SPECIAL NON-CCB FUNCTIONS

INPUTS:

R3 = ADDRESS OF FUNCTION CODE  
R4 = UNSPECIFIED CALLING PARAMETER TO DDM  
STACK CONTAINS:  
00(SP) = LINE NUMBER (ORIGINALLY IN R3)  
02(SP) = RETURN ADDRESS TO CALLING DLC

OUTPUTS TO DDM:

R3 = SUBFUNCTION CODE (WORD INDEX)  
R4 = SAME AS ON ENTRY  
R5 = LINE TABLE ADDRESS

OUTPUTS ON RETURN TO CALLER:

FUNCTION HAS BEEN PERFORMED  
R4 = UNSPECIFIED RETURN PARAMETER TO DLC FROM DDM

REGISTERS ACROSS CALL:

R2,R3,R5 PRESERVED  
R4 MAY BE MODIFIED BY DDM

.ENABL LSB  
DDCM1: .IF DF K\$\$DAS

MOV R3,\$DDFNC ; STORE FUNCTION CODE IN DATA AREA  
MOV #DDFNC,R3 ; POINT R3 AT FUNCTION CODE

.ENDC

MOV R2,-(SP) ; SAVE R2  
MOV 2(SP),R2 ; GET LINE NUMBER  
BR 22\$ ; JOIN COMMON CODE TO DISPATCH TO DDM

000140 010246  
000142 016602 000002  
000146 000452

CEDDMM CREATED BY MACRO ON 28-JUN-85 AT 18:18 PAGE 2 H 8  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL | VALUE      | REFERENCES                     |
|--------|------------|--------------------------------|
| LN.LOO | = 000003   | #5-68                          |
| LN.OAU | = 000003   | #5-68                          |
| LN.OFF | = 000001   | #5-68                          |
| LN.ON  | = 000000   | #5-68                          |
| LN.OOP | = 000004   | #5-68                          |
| LN.OPE | = 000001   | #5-68                          |
| LN.REF | = 000002   | #5-68                          |
| LN.SER | = 000002   | #5-68                          |
| LN.STA | = 000017   | #5-68                          |
| LN.SUB | = 000360   | #5-68                          |
| LN.TRI | = 000006   | #5-68                          |
| L\$SI1 | = *****    | 12-406                         |
| L.COST | = 000015   | #5-68                          |
| L.CTL  | 000012     | #5-68                          |
| L.CVA  | 177776     | #5-68                          |
| L.DDM  | 000002     | #5-68 15-577                   |
| L.DDS  | 000004     | #5-68 15-576                   |
| L.DLC  | 000003     | #5-68                          |
| L.DLM  | 000006     | #5-68                          |
| L.DLS  | 000010     | #5-68                          |
| L.FLG  | 000000     | #5-68                          |
| L.KRBA | 000016     | #5-68                          |
| L.LEN  | = 000022   | #5-68                          |
| L.MPF  | 000022     | #5-68                          |
| L.NMST | 000020     | #5-68                          |
| L.NSTA | 000014     | #5-68                          |
| L.OWNR | 000021     | #5-68                          |
| L.UNT  | 000013     | #5-68                          |
| M\$PRO | = *****    | 14-516                         |
| N\$1LN | = *****    | 10-284 15-558 15-564           |
| PR7    | = ***** GX | 12-404                         |
| PS     | = ***** GX | 12-396 *12-404 *12-414 *12-421 |
| R\$11D | = *****    | 5-67                           |
| R\$11M | = 000000   | 5-67                           |
| R\$11S | = *****    | 5-67                           |
| SF.ACT | = 000200   | #5-68                          |
| SF.ENA | = 000100   | #5-68                          |
| SF.LPB | = 000004   | #5-68                          |
| SF.MFL | = 000040   | #5-68                          |
| SF.PAC | = 000020   | #5-68                          |
| SF.REA | = 000010   | #5-68                          |
| SF.SER | = 000001   | #5-68                          |
| SF.SVC | = 000002   | #5-68                          |
| SF.UNL | = 000040   | #5-68                          |
| S.COST | = 000001   | #5-68                          |
| S.FLG  | = 000000   | #5-68                          |
| S.LEN  | = 000004   | #5-68                          |
| S.NMST | = 000002   | #5-68                          |
| S.OWNR | = 000003   | #5-68                          |
| X\$MCB | = *****    | 5-67 5-67                      |
| X\$MDC | = 000001   | #4-2 5-2 5-3 10-272            |
| X\$NDM | = *****    | 5-1 5-3 7-116                  |

```

293
294
295 .ENDC
296 BIT #LF.MDC,@(R2)+ ; DOES LINE NEED MODEM CONTROL?
297 BEQ 20$; NO - JOIN COMMON CODE TO DISPATCH TO DDM
298 MOV R5,-(SP) ; SAVE REGISTER
299 MOV R3,-(SP) ; SAVE ADDRESS OF FUNCTION CODE
300 CLR R2 ; GET AUXILIARY PROCESS PDX INDEX (ALWAYS 0)
301 MOV @SPDVTAR5 ; GET ADDRESS OF AUXILIARY PROCESS' PDV
302 ; THAT WORKS SINCE IT'S PDV INDEX = 0
303 MOV Z.DAT(R5),R5 ; GET ADDRESS OF DATA BASE DESCRIPTOR BLOCK
304 CALL $PDDSP ; DISPATCH TO MODEM CONTROLLER
305 MOV (SP)+,R3 ; RESTORE R3
306
307 MOVB C.LIN(R4),R2 ; OBTAIN LINE NUMBER (NO SIGN EXTENSION)
308 BIS #100000,C.LIN(R4) ; INDICATE MODEM CONTROL REQUEST
309 CALL DDMDSP ; DISPATCH TO THE DEVICE DRIVER (DDM)
310 BCS 25$; REQUEST WILL COMPLETE ASYNCHRONOUSLY
311 CALL $CCBRT ; OTHERWISE RETURN CCB TO POOL
312 CLR R4 ; MODEM CONTROLLER WILL RETURN ASYNCHRONOUS COMPLETION
313 SEC R4 ; SET C-BIT, LINE IS MODEM CONTROLLED
314 BR 25$; JOIN COMMON CODE
315 .ENDC

```

\*\*FILE\*\*ID\*\*CEDLC

```

CCCCCCCC EEEEEEEEE DDDDDDDD LL CCCCCCCC
CCCCCCCC EEEEEEEEE DDDDDDDD LL CCCCCCCC
CC EE DD DD CC
CC EE DD DD CC
CC EE DD DD CC
CC EE DD DD CC
CC EEEEEEEE DD DD CC
CC EEEEEEEE DD DD CC
CC EE DD DD CC
CC EE DD DD CC
CC EE DD DD CC
CC EE DD DD CC
CCCCCCCC EEEEEEEEE DDDDDDDD LLLLLLLLLL CCCCCCCC
CCCCCCCC EEEEEEEEE DDDDDDDD LLLLLLLLLL CCCCCCCC

```

```

....
....
....
....

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```

CEDLC      CREATED BY    MACRO    ON 28-JUN-85 AT 18:19

PAGE 4      H 11

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|         |       |       |
|---------|-------|-------|
| CALL    | 6-140 | 7-208 |
| CALLR   | #4-61 |       |
| CCBDF\$ | #4-60 | 4-62  |
| PDVDF\$ | #4-60 | 4-63  |
| RESRG   | #4-59 |       |
| RETURN  | 5-112 | 7-211 |
| SAVRG   | #4-59 |       |
| SLTDF\$ | #4-60 | 4-64  |

CELLC      CREATED BY    MACRO    ON 28-JUN-85 AT 18:19

PAGE 3    H 12

SYMBOL CROSS REFERENCE

CREF    04.00

| SYMBOL  | VALUE       | REFERENCES       |
|---------|-------------|------------------|
| \$LLCRS | 000074 RG   | #6-168           |
| \$NMCRS | 000074 RG   | #6-174           |
| \$PDDSP | =        GX | 7-227            |
| \$PDQU1 | =        GX | 6-178            |
| \$PDSPL | =        GX | 5-121            |
| \$PDVTA | =        GX | 5-112      7-223 |
| \$STDLC | =        GX | 5-120            |



H 13

```

56 .SBTTL LOG NETWORK EVENT
57
58
59 ***$CELOG-LOG NETWORK EVENT
60
61 INPUTS:
62 R0 - EVENT CLASS AND TYPE
63 R1 - EVENT CONTROL WORD:
64 LOW BYTE:
65 BIT 0 1 => R4 IS ADDRESS OF A CCB
66 0 => R4 IS A DATA AREA POINTER
67
68 1 1 => LINE-ID IS SLN & STATION
69 0 => LINE-ID IS PDV & CHANNEL
70
71 2 1 => EVENT IS ASSOCIATED WITH A LINE
72 0 => EVENT IS NOT ASSOCIATED WITH A LINE
73
74 3 1 => EVENT IS ASSOCIATED WITH A REMOTE NODE
75 0 => EVENT IS NOT ASSOCIATED WITH A REMOTE NODE
76
77 4 1 => EVENT IS ASSOCIATED WITH A CIRCUIT
78 0 => EVENT IS NOT ASSOCIATED WITH A CIRCUIT
79
80 5 1 => EVENT IS ASSOCIATED WITH A MODULE
81 0 => EVENT IS NOT ASSOCIATED WITH A MODULE
82
83 6 1 => USE LINE-ID FROM CCB
84 0 => USE LINE-ID FROM EVENT DESCRIPTOR BLOCK
85
86 HIGH BYTE:
87 # OF BYTES OF DATA TO COPY
88
89 R3 - POINTER TO EVENT DESCRIPTOR BLOCK:
90
91 WORD 1 LINE-ID (SLN & STATION OR PDV & CHANNEL)
92 2-4 EVENT DEPENDENT DATA
93 5 NODE ADDRESS
94 6 MODULE ID (CODED VALUE)
95 7 PORT #
96 8 LOGICAL CHANNEL #
97
98 R4 - POINTER TO CCB OR DATA AREA POINTER
99
100 REGISTERS MODIFIED:
101 R0, R1, R2, R3, R5
102
103
104 $CELOG::IF DF N$$EVL
105
106 000000 016702 000000G MOV $LGPDV,R2 ; GET PDV INDEX OF LOGGING PROCESS
107 000004 001417 BEQ 10$; IF EQ, NO LOGGING
108
109 000006 016746 000000G MOV $CMPDV,-(SP) ; SAVE CALLING PROCESS PDV INDEX
110 000012 010346 MOV R3,-(SF) ; SAVE ADDRESS OF EVENT DESCRIPTOR BLOCK
111 000014 010146 MOV R1,-(SP) ; AND EVENT CONTROL WORD
112 000016 010601 MOV SP,R1 ; SET UP POINTER TO CONTROL BLOCK

```

I 13

```

217 .SBTTL $PDDSP - DISPATCH TO A PROCESS
218
219
220 ;+
221 ;**-$PDDSP-DISPATCH TO A PROCESS
222 ;
223 ;Inputs:
224 ;
225 ; R2 = PDV Index (word index)
226 ; R3 = Address of function code
227 ; R4 = Address of a CCB (optional)
228 ; R5 = Address of a process line table (optional)
229 ;
230 ;Outputs to the Process:
231 ;
232 ; Based on the PDV Index in R2, the process is entered at the
233 ; entry point specified by the function code pointed to by R3
234 ; with:
235 ; R3 = Subfunction code (word index)
236 ; R4 = Address of a CCB (optional)
237 ; R5 = Address of process line table (optional)
238 ;
239 ; R2 through R5 need not be preserved by the called process
240 ;-
241
242 000076
243 000076 016746 000000G
244 000102 010267 000000G
245 000106 066702 000000G
246 000112 011202
247 000114 016746 000000G
248 000120 016746 000000G
249
250
251
252
253
254
255 000124 012267 000000G
256 000130 011246
257 000132 112302
258 000134 062602
259 000136 032712 000001
260 000142 001011
261 000144 112303
262 000146
263
264
265
266
267
268
269 000150 012667 000000G
270 000154 012667 000000G
271 000160 012667 000000G
272 000164
273

```

```

$PDDSP::
MOV $CMPDV,-(SP) ; ref label
MOV R2,$CMPDV ; Save Current Process PDV Index
ADD $PDVTA,R2 ; Load new Process PDV Index
MOV (R2),R2 ; Point into PDV Index Table
MOV KISAR6,-(SP) ; Get PDV Address
MOV KISAR5,-(SP) ; Save current Kernel APR 6
; Save current process mapping

; IF DF K$$DAS
MOV KINAR6,-(SP) ; Save instruction APR'S
MOV KINAR5,-(SP)
MOV (R2),KINAR5 ; Map process in both I and D space
.ENDC ; DF K$$DAS

MOV (R2)+,KISAR5 ; Map to process
MOV (R2)-,(SP) ; Save address of dispatch table
MOVB (R3)+,R2 ; Get function code (no sign extend)
ADD (SP)+,R2 ; Form address of entry in dispatch table
BIT #1,(R2) ; Is this a dispatch to an odd address?
BNE 10$; If NE, yes! - bad dispatch.
MOVB (R3)+,R3 ; Set up subfunction code
CALL @ (R2)+ ; Call process

; IF DF K$$DAS
MOV (SP)+,KINAR5 ; Restore instruction APRs
MOV (SP)+,KINAR6
.ENDC ; DF K$$DAS

MOV (SP)+,KISAR5 ; Restore previous process mapping
MOV (SP)+,KISAR6 ; Restore previous APR 6 contents
MOV (SP)+,$CMPDV ; Restore previous process PDV index
RETURN ; And return to caller

```

```
58 ; MACRO LIBRARY CALLS
59 ;
60 .MCALL INHIB$,ENABL$,SAVRG,RESRG
61 .MCALL CCBDF$,PDVDF$,CLKDF$
62 .MCALL CALLR ; AVOID SYSTEM DEPENDENCY
63 000000 ; DEFINE THE CCB OFFSETS
64 000000 ; DEFINE THE PDV OFFSETS
65 000000 ; DEFINE CLOCK BLOCK OFFSETS
66
```

```

537 .SBTTL $MPSAV - BYPASS CACHE WITH SAVE AND RESTORE
538
539 ;+
540 **-$MPSAV-BYPASS CACHE WITH SAVE AND RESTORE
541
542 THIS ROUTINE WILL SAVE THE CURRENT STATE OF THE CACHE AND BYPASS
543 IT BEFORE CO-CALLING THE CALLER. ON RETURN IT WILL RESTORE THE
544 INITIAL STATE OF THE CACHE.
545
546 $MPSAV::CACHE$ SAVE ; SAVE CURRENT STATE OF CACHE
547
548 CALL @2(SP) ; CO-CALL THE CALLER
549 MOV (SP)+,2(SP) ; OVERWRITE RETURN LINK
550
551 CACHE$ UNSAVE ; RESTORE THE CACHE
552 RETURN
553
554 .ENDC ; DF M$$PRO
555
556 000001 ;
557 .END

```

```

137 .SBTTL $RDBG - Get a Receive Data Buffer
138 +
139 **-$RDBG- Get a Receive Data Buffer
140
141 This subroutine is called to allocate a fixed length Receive
142 Data Buffer.
143
144 Inputs:
145 None.
146
147 Outputs:
148 R4 contains the address of allocated buffer's CCB
149 C-Bit is CLEAR if the CCB/RDB was successfully allocated
150 C-Bit is SET and R4=0 if the allocation failed
151
152 Note:
153 The caller's APR6 is preserved across this call.
154 -
155
156 $RDBG::SAVMAP : Save current mapping
157 SAVRG <R2,R3,R5> : Save some registers
158 CLC : Assume successful
159 INHIB$::: Inhibit Interrupts (and save C-Bit)
160
161 .IF DF M$$PRO
162 CALL $MPLCK ::: Lock access to CommExec resources
163 .ENDC
164
165 CALL CCBGT ::: Allocate a CCB
166 BCS 10$::: If CS, None available
167
168 MOV #CF.LB,C.FLG(R4) ::: Set-up last buffer indicator
169 MOVB #CB.RDB,C.BID(R4) ::: and buffer ID fields in CCB
170 MOV $RDBSZ,C.CNT(R4) ::: Load size of RDB/LDB
171 MOV $RDBSZ,C.CNT2(R4)
172 MOV #RDBLH,R5 ::: Point to listhead in allocation ctrl block
173 CALL BUFGT ::: Allocate a Large Data Buffer
174 BCC 10$::: If CC, Successful
175
176 CALL CCBRT ::: Else, release the CCB
177 CLR R4 ::: No CCB available
178 10$:
179 .IF DF M$$PRO
180 CALL @($P)+ ::: Co-Routine return to unlock resources
181 .ENDC
182
183 ENABL$: Enable Interrupts (and load final C-Bit)
184 RESRG <R5,R3,R2> : Restore registers
185 RESMAP : Restore mapping
186 RETURN : Return to caller

```

CEBUF - CEX BUFFER MANAGEMENT R MACRO V05.03b Friday 28-Jun-85 18:16<sup>1 2</sup> Page 20  
BUFRT - General Buffer Deallocation Routine

627

000001

.END

```

137 .SBTTL $RDBGT - Get a Receive Data Buffer
138 ;+
139 ;**-$RDBGT- Get a Receive Data Buffer
140 ;
141 ; This subroutine is called to allocate a fixed length Receive
142 ; Data Buffer.
143 ;
144 ; Inputs:
145 ; None.
146 ;
147 ; Outputs:
148 ; R4 contains the address of allocated buffer's CCB
149 ; C-Bit is CLEAR if the CCB/RDB was successfully allocated
150 ; C-Bit is SET and R4=0 if the allocation failed
151 ;
152 ; Note:
153 ; The caller's APR6 is preserved across this call.
154 ;-
155
156 000044 $RDBGT::SAVMAP ; Save current mapping
157 000050 SAVRG <R2,R3,R5> ; Save some registers
158 000056 000241 CLC ; Assume successful
159 000060 INHIB$;;; Inhibit Interrupts (and save C-Bit)
160
161 .IF DF M$$PRO
162 CALL $MPLCK ;;; Lock access to CommExec resources
163 .ENDC
164
165 000072 CALL CCBGT ;;; Allocate a CCB
166 000076 103424 BCS 10$;;; If CS, None available
167
168 000100 012764 100000 000022 MOV #CF.LB,C.FLG(R4) ;;; Set-up last buffer indicator
169 000106 112764 000004 000003 MOV #CB.RDB,C.BID(R4) ;;; and buffer ID fields in CCB
170 000114 016764 000000G 000020 MOV $RDBSZ,C.CNT(R4) ;;; Load size of RDB/LDB
171 000122 016764 000000G 000030 MOV $RDBSZ,C.CNT2(R4)
172 000130 012705 000000G MOV #$RDBLH,R5 ;;; Point to listhead in allocation ctrl block
173 000134 CALL BUFGT ;;; Allocate a Large Data Buffer
174 000140 103003 BCC 10$;;; If CC, Successful
175
176 000142 CALL CCBRT ;;; Else, release the CCB
177 000146 005004 CLR R4 ;;; No CCB available
178 000150 10$: .IF DF M$$PRO
179 .CALL a(SP)+ ;;; Co-Routine return to unlock resources
180 .ENDC
181
182 000150 ENABL$; Enable Interrupts (and load final C-Bit)
183 000154 RESRG <R5,R3,R2> ; Restore registers
184 000162 RESMAP ; Restore mapping
185 000166 RETURN ; Return to caller

```

CEBUF1 - CEX BUFFER MANAGEMENT MACRO V05.03b Friday 28-Jun-85 18:17<sup>J 4</sup> Page 21  
BUFRT - General Buffer Deallocation Routine

627

000001

.END



CEDDM MACRO V05.03b Friday 28-Jun-85 18:17 Page 7  
 \$DD?? - DLC TO DDM REQUESTS

|            |                |                   |                             |
|------------|----------------|-------------------|-----------------------------|
| 174 000000 | \$DDXME::DDFDF | FC.XME,SUB        | ; TRANSMIT ENABLE           |
| 175 000006 | \$DDRCE::DDFDF | FC.RCE,SUB        | ; RECEIVE ENABLE            |
| 176        |                |                   |                             |
| 177 000014 | \$DDMAN::DDFDF | FC.MAN,SUB        | ; NETWORK MANAGEMENT        |
| 178        |                |                   |                             |
| 179 000022 | \$DDXKL::DDFDF | FC.KIL+FS.XKL,CCB | ; TRANSMIT KILL             |
| 180 000030 | \$DDCRA::DDFDF | FC.KIL+FS.CRA,CCB | ; RECEIVE KILL              |
| 181 000036 | \$DDKIL::DDFDF | FC.KIL+FS.KIL,CCB | ; RECEIVE AND TRANSMIT KILL |
| 182 000044 | \$DDSTR::DDFDF | FC.CTL+FS.STR,CCB | ; START                     |
| 183 000052 | \$DDSTP::DDFDF | FC.CTL+FS.STP,CCB | ; STOP                      |
| 184 000060 | \$DDGET::DDFDF | FC.CTL+FS.GET,CCB | ; GET CHARACTERISTICS       |
| 185 000066 | \$DDSET::DDFDF | FC.CTL+FS.SET,CCB | ; SET CHARACTERISTICS       |
| 186        |                |                   |                             |
| 187 000074 | \$DDRNG::DDFDF | FC.CTL+FS.RNG,MDC | ; LOOK FOR RING             |
| 188 000102 | \$DDENB::DDFDF | FC.CTL+FS.ENB,MDC | ; ENABLE LINE               |
| 189 000110 | \$DDDIS::DDFDF | FC.CTL+FS.DIS,MDC | ; DISABLE LINE              |
| 190        |                |                   |                             |
| 191 000116 | \$DDXON::DDFDF | FC.CTL+FS.XON     | ; XON                       |
| 192 000124 | \$DDXOF::DDFDF | FC.CTL+FS.XOF     | ; XOFF                      |
| 193 000132 | \$DDMSN::DDFDF | FC.CTL+FS.MSN     | ; SENSE MODEM STATUS        |
| 194        |                |                   |                             |

CEDDM      CREATED BY MACRO ON 28-JUN-85 AT 18:18      PAGE 1      I 6

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE    | REFERENCES |
|--------|----------|------------|
| C.FNC  | 000010   | *10-350    |
| C.LIN  | 000006   | *10-361    |
| C.STS  | 000012   | *12-456    |
| DDCM1  | 000140 R | *13-497    |
| DDCM2  | 000150 R | *13-500    |
| DDCM3  | 000150 P | *13-503    |
| DDCM4  | 000156 R | *13-510    |
| DDMDSP | 000206 R |            |
| FC.CCP | = 000020 |            |
| FC.CTL | = 000006 |            |
| FC.KCP | = 000016 |            |
| FC.KIL | = 000004 |            |
| FC.MAN | = 000024 |            |
| FC.RCE | = 000002 |            |
| FC.RCP | = 000014 |            |
| FC.XCP | = 000012 |            |
| FC.XME | = 000000 |            |
| FS.AST | = 000000 |            |
| FS.CRA | = 001000 |            |
| FS.DIS | = 013000 |            |
| FS.ENB | = 012000 |            |
| FS.GET | = 006000 |            |
| FS.KIL | = 000000 |            |
| FS.MSN | = 014000 |            |
| FS.RNG | = 011000 |            |
| FS.SET | = 005000 |            |
| FS.STP | = 002000 |            |
| FS.STR | = 001000 |            |
| FS.XKL | = 002000 |            |
| FS.XOF | = 010000 |            |
| FS.XON | = 007000 |            |
| ISSAS  | = *****  |            |
| KSSDAS | = *****  |            |
| LF.ACT | = 100000 |            |
| LF.BRO | = 000400 |            |
| LF.BWT | = 000007 |            |
| LF.ENA | = 002000 |            |
| LF.LPB | = 001000 |            |
| LF.MDC | = 000100 |            |
| LF.MFL | = 004000 |            |
| LF.MTP | = 000020 |            |
| LF.PAC | = 000200 |            |
| LF.RDY | = 040000 |            |
| LF.REA | = 010000 |            |
| LF.SER | = 000040 |            |
| LF.TIM | = 000010 |            |
| LF.UNL | = 020000 |            |
| LF.X2P | = 000000 |            |
| LN.CLO | = 000000 |            |
| LN.DUM | = 000005 |            |
| LN.LOA | = 000004 |            |

```

236 .SBTTL DDCM2 - COMMON PROCESS FOR MODEM CONTROL FUNCTIONS
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271 000150
272
273
274
275
276
277
278
279
280
281 000150 011364 000010
282 000154 010246
283
284
285
286
287
288
289
290 000156 116402 000006
291 000162 006302
292 000164 066702 000000G

 **--DDCM2 - COMMON PROCESS FOR MODEM CONTROL FUNCTIONS
 IF THE SPECIFIED LINE HAS MODEM CONTROL ENABLED THE
 FUNCTION REQUEST IS DERAILED TO THE MODEM CONTROLLER.
 ON RETURN FROM THE MODEM CONTROLLER THE REQUEST IS
 ROUTED TO THE PROPER DDM AS USUAL. IF THE LINE IS
 HARDWIRED THE REQUEST IS ROUTED DIRECTLY TO THE DDM.
 INPUTS:
 R3 = ADDRESS OF FUNCTION CODE
 R4 = ADDRESS OF CCB
 STACK CONTAINS:
 00(SP) = CALLER'S R3
 02(SP) = RETURN ADDRESS TO CALLING DLC

 OUTPUTS TO MODEM CONTROLLER:
 R3 = SUBFUNCTION CODE
 R5 = LINE TABLE ADDRESS

 OUTPUTS TO DDM:
 R3 = SUBFUNCTION CODE
 R4 = SAME AS ON ENTRY
 R5 = LINE TABLE ADDRESS

 OUTPUTS ON RETURN TO CALLER:
 C-BIT CLEAR - FUNCTION PERFORMED SYNCHRONOUSLY
 C-BIT SET - WAIT FOR ASYNCHRONOUS REPORT OF FUNCTION COMPLETION

 REGISTERS ACROSS CALL:
 R2,R3,R5 - PRESERVED
 R4 - MODIFIED

DDCM2:
 .IF DF X$$MDC
 .IF DF K$$DAS
 MOV R3,$DDFNC ; STORE THE FUNCTION CODE IN DATA AREA
 MOV #DDFNC,R3 ; POINT R3 AT FUNCTION CODE
 .ENDC
 MOV (R3),C.FNC(R4) ; PUT FUNCTION CODE IN CCB
 MOV R2,-(SP) ; SAVE R2
 .IF DF N$$1LN
 MOV $SLTMA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE
 .IFF
 MOVB C,LIN(R4),R2 ; GET SYSTEM LINE NUMBER
 ASL R2 ; FORM WORD INDEX
 ADD $SLTMA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE

```

| SYMBOL  | VALUE      | REFERENCES   |
|---------|------------|--------------|
| ZF.COU  | = 001000   | #5-67        |
| ZF.DDM  | = 000001   | #5-67        |
| ZF.DIA  | = 004000   | #5-67        |
| ZF.DLC  | = 000002   | #5-67        |
| ZF.DVP  | = 100000   | #5-67 12-401 |
| ZF.INI  | = 040000   | #5-67        |
| ZF.KMX  | = 000020   | #5-67        |
| ZF.LLC  | = 000004   | #5-67        |
| ZF.LMC  | = 000100   | #5-67        |
| ZF.MAN  | = 020000   | #5-67        |
| ZF.MFL  | = 000010   | #5-67        |
| ZF.MTM  | = 000400   | #5-67        |
| ZF.MUX  | = 000040   | #5-67        |
| ZF.PSE  | = 002000   | #5-67        |
| ZF.SLI  | = 010000   | #5-67        |
| ZF.TIM  | = 000200   | #5-67        |
| ZF.X3P  | = 000000   | #5-67        |
| ZS.ASN  | = 100000   | #5-67        |
| ZS.BSY  | = 140000   | #5-67        |
| Z.AVL   | 000014     | #5-67        |
| Z.DAT   | 000016     | #5-67 10-303 |
| Z.DSP   | 000000     | #5-67 5-67   |
| Z.FLG   | 000010     | #5-67 12-401 |
| Z.LEN   | = 000016   | #5-67        |
| Z.LLN   | 000006     | #5-67        |
| Z.MAP   | 000020     | #5-67        |
| Z.NAM   | 000004     | #5-67        |
| Z.PCB   | 000012     | #5-67        |
| Z.SCH   | 000007     | #5-67 12-414 |
| \$CCBGT | = ***** GX | 13-453       |
| \$CCBRT | = ***** GX | 10-311       |
| \$DDAST | 000400 RG  | #13-452      |
| \$DDCCP | 000460 RG  | #14-506      |
| \$DDCRA | 000030 RG  | #8-180       |
| \$DDDIS | 000110 RG  | #8-189       |
| \$DDENB | 000102 RG  | #8-188       |
| \$DDGET | 000060 RG  | #8-184       |
| \$DDKCP | 000450 RG  | #14-503      |
| \$DDKIL | 000036 RG  | #8-181       |
| \$DDMAN | 000014 RG  | #8-177       |
| \$DDMSN | 000132 RG  | #8-193       |
| \$DDRCE | 000006 RG  | #8-175       |
| \$DDRCF | 000440 RG  | #14-500      |
| \$DDRNG | 000074 RG  | #8-187       |
| \$DDSET | 000066 RG  | #8-185       |
| \$DDSPC | 000500 RG  | #14-513      |
| \$DDSTP | 000052 RG  | #8-183       |
| \$DDSTR | 000044 RG  | #8-182       |
| \$DDXKI | 000022 RG  | #8-179       |
| \$DDXME | 000000 RG  | #8-174       |
| \$DDXMP | 000430 RG  | #14-497      |
| \$DDXOF | 000124 RG  | #8-192       |

```

317 .SBTTL DDCM3 - COMMON PROCESS FOR FUNCTIONS WITH CCB
318 .SBTTL DDCM4 - COMMON PROCESS FOR FUNCTIONS WITH CCB
319
320 ;+
321 **-DDCM3 - COMMON PROCESS FOR FUNCTIONS WITH CCB
322 **-DDCM4 - COMMON PROCESS FOR FUNCTIONS WITH CCB
323
324 INPUTS:
325 R3 = ADDRESS OF FUNCTION AND SUBFUNCTION CODE (DDCM3)
326 R3 = ADDRESS OF FUNCTION CODE ONLY (DDCM4)
327 R4 = ADDRESS OF CCB
328 STACK CONTAINS:
329 00(SP) = CALLERS R3
330 02(SP) = RETURN ADDRESS TO CALLING DLC
331
332 OUTPUTS TO DDM:
333 R3 = SUBFUNCTION CODE (WORD INDEX)
334 R4 = ADDRESS OF CCB
335 R5 = ADDRESS OF LINE TABLE
336
337 REGISTERS ACROSS CALL:
338 R2,R3,R5 - PRESERVED
339 R4 - MAY BE MODIFIED
340 -
341
342 DDCM3:
343 .IF DF K$$DAS
344
345 MOV R3,$DDFNC ; STORE FUNCTION CODE IN DATA AREA
346 MOV #DDDFNC,R3 ; POINT R3 AT FUNCTION CODE
347
348 .IFTF ; DF K$$DAS
349
350 MOV (R3),C.FNC(R4) ; PUT FUNCTION AND SUBFUNCTION IN CCB
351 BR 10$; JOIN COMMON CODE
352
353 DDCM4:
354 .IFT ; DF K$$DAS
355
356 MOV R3,$DDFNC ; STORE FUNCTION CODE IN DATA AREA
357 MOV #DDDFNC,R3 ; POINT R3 AT FUNCTION CODE
358
359 .ENDC ; DF K$$DAS
360
361 MOVVB (R3),C.FNC(R4) ; PUT FUNCTION CODE IN CCB
362
363 10$: MOV R2,-(SP) ; SAVE R2
364
365 20$: MOVVB C.LIN(R4),R2 ; GET SLN
366
367 22$: MOV R5,-(SP) ; SAVE R5
368 CALL DDMDSP ; DISPATCH TO THE DEVICE DRIVER (DDM)
369 25$: MOV (SP)+,R5 ; RESTORE R5
370 MOV (SP)+,R2 ; RESTORE R2
371 MOV (SP)+,R3 ; RESTORE R3
372 RETURN ; RETURN TO CALLING DLC
373

```

|    |     |                                                 |
|----|-----|-------------------------------------------------|
| 5- | 67  | \$STDLC - SET UP DLC PARAMETERS FOR A GIVEN SLN |
| 6- | 115 | \$ASCMP - ASYNCHRONOUS COMPLETION TO LLC LEVEL  |
| 7- | 147 | \$XMCMP - TRANSMIT COMPLETE TO LLC LEVEL        |
| 7- | 148 | \$CTCMP - CONTROL COMPLETE TO LLC LEVEL         |
| 7- | 149 | \$RCCMP - RECEIVE COMPLETE TO LLC LEVEL         |
| 7- | 150 | \$KLCMP - KILL COMPLETE TO LLC LEVEL            |

\*\*FILE\*\*ID\*\*CELLC

```

CCCCCCCC EEEEEEEEE LL LL CCCCCCCC
CCCCCCCC EEEEEEEEE LL LL CCCCCCCC
CC EE LL LL CC
CC EE LL LL CC
CC EE LL LL CC
CC EE LL LL CC
CC EEEEEEEE LI LL CC
CC EEEEEEEE LL LL CC
CC EE LL LL CC
CC EE LL LL CC
CC EE LL LL CC
CC EE LL LL CC
CCCCCCCC EEEEEEEEE LLLLLLLLL LLLLLLLLL CCCCCCCC
CCCCCCCC EEEEEEEEE LLLLLLLLL LLLLLLLLL CCCCCCCC

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```

```

....
....
....
....

```

CELLC      CREATED BY    MACRO    ON 28-JUN-85 AT 18:19      PAGE 4      I 12  
MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES                             |
|------------|----------------------------------------|
| CALL       | 5-120      5-121      6-178      7-227 |
| CALLR      | #4-59                                  |
| CCBDF\$    | #4-58      4-60                        |
| MTPS       | 5-122                                  |
| PDVDF\$    | #4-58      4-61                        |
| RESRG      | #4-58      5-123      6-179      7-229 |
| RETURN     | 5-124      6-180      7-230            |
| SAVRG      | #4-58      5-110      6-174      7-221 |
| SLTDF\$    | #4-58      4-62                        |



```

113
114 000020 SAVRG <R4> ; PRESERVE CCB ADDRESS
115 000022 016705 000000G MOV $LGDDB,R5 ; SET ADDRESS OF LOGGING DATABASE
116 000026 012703 000000G MOV #CELFN,R3 ; SET UP DISPATCH FUNCTION
117 000032 CALL $PDDSP ; AND DISPATCH TO LOGGING PROCESS
118 000036 RESRG <R4> ; RESTORE CCB ADDRESS
119
120 000040 022626 CMP (SP)+,(SP)+ ; CLEAN UP THE STACK
121 000042 005726 TST (SP)+ ; ...
122
123 .ENDC
124
125 000044 10$: RETURN ; AND RETURN
126
127 000001 .END

```

```
274 ; at crash:
275 ; R2 -> computed address of dispatch table entry.
276 ; R3-1 -> byte of function code.
277 ; R3 -> byte of subfunction code.
278 ; R4 = optional CCB address.
279 ; R5 = optional line table address.
280
281 000166 000004 10$: IOT ; Crash on odd dispatch.
```

68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99

.SBTTL \$CMQIN - QUEUE A CHAIN OF CCBS TO A LIST

\*\*\*\$CMQIN - QUEUE A CHAIN OF CCBS TO A LIST

THIS SUBROUTINE IS CALLED TO ADD A CCB OR A CHAIN OF CCBS TO  
 THE END OF A SINGLE LINKED LIST.

INPUTS:

R3 = ADDRESS OF TWO WORD LISTHEAD  
 R4 = ADDRESS OF FIRST CCB IN CHAIN

OUTPUTS:

THE CCB IS QUEUED TO THE END OF THE LIST

REGISTERS MODIFIED:

NONE

```

$CMQIN::SAVRG <R5,R4> ; SAVE REGISTERS
10$: MOV R4,R5 ; COPY CCB ADDRESS
 BIC #CS.LST,C.STS(R5) ; CLEAR LAST CCB IN CHAIN BIT
 MOV (R5),R4 ; GET ADDRESS OF NEXT CCB IN CHAIN
 BNE 10$; LOOP TILL END OF CHAIN
 BIS #CS.LST,C.STS(R5) ; SET LAST CCB IN CHAIN BIT
 MOV (SP),a2(R3) ; LINK CHAIN TO END OF QUEUE
 MOV R5,2(R3) ; UPDATE LISTHEAD POINTER TO LAST CCB IN QUEUE
 RESRG <R4,R5> ; RESTORE REGISTERS
 RETURN ; RETURN

```

```

000000
000004 010405
000006 042765 040000 000012
000014 011504
000016 001372
000020 052765 040000 000012
000026 011673 000002
000032 010563 000002
000036
000042

```

CESUB MACRO V05.03b Friday 28-Jun-85 18:20 Page 16-1  
Symbol table

|                |                |                 |                 |                   |
|----------------|----------------|-----------------|-----------------|-------------------|
| ASSCHK= 000000 | CS.DIS= 000040 | C.SYTK= 000010  | FS.SFC= 005000  | SS\$YSZ= 007600   |
| ASSCP= 000000  | CS.ENA= 000001 | C.TCB= 000004   | FS.SFR= 006000  | SS\$KMG= 000000   |
| ASSPRI= 000000 | CS.ENB= 000020 | C.TIM= 000006   | FS.SFS= 004000  | SS\$MIN= 000000   |
| ASSTRP= 000000 | CS.ERR= 100000 | C.UIC= 000016   | FS.SPW= 040000  | VS\$CTR= 001000   |
| CB.CCB= 000002 | CS.FTL= 001000 | C.URM= 177776   | FS.STM= 000000  | XSSDBT= 000000    |
| CB.DDM= 000040 | CS.HCR= 000001 | C.XACP= 000004  | FS.STP= 002000  | ZF.COU= 001000    |
| CB.DLC= 000020 | CS.HFE= 002000 | C.XID= 000035   | FS.STR= 001000  | ZF.DDM= 000001    |
| CB.RDB= 000004 | CS.LST= 040000 | C.XLEN= 000044  | FS.TRM= 003000  | ZF.DIA= 004000    |
| CB.SDB= 000010 | CS.MTL= 004000 | C.XPLI= 000040  | FS.WLB= 001000  | ZF.DLC= 000002    |
| CB.SLI= 000100 | CS.RNG= 000010 | C.XPT= 000034   | FS.XKL= 002000  | ZF.DVP= 100000    |
| CB.XLB= 000001 | CS.ROV= 000004 | C.XSVC= 000042  | FS.XOF= 010000  | ZF.INI= 040000    |
| CC.LLC= 000200 | CS.RSN= 010000 | C.XTC= 000037   | FS.XON= 007000  | ZF.KMX= 000020    |
| CE.ABO= 100362 | CS.SHU= 000001 | C.X25= 000036   | FS.ZER= 002000  | ZF.LLC= 000004    |
| CE.DAO= 100346 | CS.SID= 000002 | D\$BBUG= 177514 | F\$LLVL= 000001 | ZF.LMC= 000100    |
| CE.DIS= 100366 | CS.STR= 000004 | D\$BISK= 000000 | G\$STPP= 000000 | ZF.MAN= 020000    |
| CE.ERR= 100370 | CS.SUC= 000001 | D\$LL11= 000001 | G\$STSS= 000000 | ZF.MFL= 000010    |
| CE.ILN= 100350 | CS.TMO= 020000 | D\$SYNC= 000000 | G\$STTK= 000000 | ZF.MTM= 000400    |
| CE.LTO= 100356 | CS.XUR= 000004 | D\$SYNM= 000000 | G\$SWRD= 000000 | ZF.MUX= 000040    |
| CE.MOP= 100372 | C\$CKP= 000000 | E\$XPR= 000000  | I\$RAR= 000000  | ZF.PSE= 002000    |
| CE.NTE= 100361 | C\$GHE= 000400 | FC.CCP= 000020  | I\$SRDN= 000000 | ZF.SLI= 010000    |
| CE.RTE= 100376 | C\$RSH= 177564 | FC.CTL= 000006  | KISAR5= *****   | ZF.TIM= 000200    |
| CE.SRC= 100364 | C.ADD= 000034  | FC.KCP= 000016  | KISAR6= *****   | ZF.X3P= 000000    |
| CE.STP= 100352 | C.AR5= 000014  | FC.KIL= 000004  | K\$CNT= 177546  | ZS.ASN= 100000    |
| CE.TME= 100354 | C.AST= 000012  | FC.MAN= 000024  | K\$CSR= 177546  | ZS.BSY= 140000    |
| CF.TMO= 100374 | C.BID= 000003  | FC.MLD= 000026  | K\$LDC= 000000  | Z.AVL= 000014     |
| CE.UNS= 100344 | C.BUF= 000014  | FC.PCT= 000030  | K\$TPS= 000074  | Z.DAT= 000016     |
| CF.CHN= 000001 | C.BUF1= 000014 | FC.PWR= 000022  | LD\$LP= 000000  | Z.DSP= 000000     |
| CF.EOM= 000004 | C.BUF2= 000024 | FC.RCE= 000002  | L\$ASG= 000000  | Z.FLG= 000010     |
| CF.HDR= 000020 | C.CNT= 000020  | FC.RCP= 000014  | L\$DRV= 000000  | Z.LEN= 000016     |
| CF.LB= 100000  | C.CNT1= 000020 | FC.TIM= 000010  | L\$P11= 000001  | Z.LLN= 000006     |
| CF.LIN= 000002 | C.CNT2= 000030 | FC.XCP= 000012  | L\$11R= 000000  | Z.MAP= 000020     |
| CF.SOM= 000010 | C.CSTP= 000012 | FC.XME= 000000  | M\$SCRB= 000124 | Z.NAM= 000004     |
| CF.SYN= 000040 | C.DST= 000016  | FS.AST= 000000  | M\$SCRX= 000000 | Z.PCB= 000012     |
| CF.TRN= 000100 | C.EFN= 000003  | FS.CIB= 002000  | M\$FCS= 000000  | Z.SCH= 000007     |
| CM.CIR= 000002 | C.FLG= 000022  | FS.CRA= 001000  | M\$MGE= 000000  | \$CALLX 000316RG  |
| CM.FMT= 100000 | C.FLG1= 000022 | FS.DIS= 013000  | M\$NET= 000000  | \$CEACC 000424RG  |
| CM.HRD= 000002 | C.FLG2= 000032 | FS.DVC= 001000  | M\$QVR= 000000  | \$CECAC 000432RG  |
| CM.LIN= 000000 | C.FNC= 000010  | FS.ENB= 012000  | N\$ACC= 000001  | \$CEDIV 000576RG  |
| CM.LOO= 000001 | C.LGTH= 000020 | FS.EXI= 001000  | N\$BUF= 000001  | \$CEMUL 000572RG  |
| CM.XLO= 000004 | C.LIN= 000006  | FS.GE1= 006000  | N\$LDV= 000001  | \$CMPDV= ***** GX |
| CP.DCF= 000040 | C.LNK= 000000  | FS.HLT= 000000  | N\$MCP= 000001  | \$CMQIN 000000RG  |
| CP.HDL= 000007 | C.MOD= 000011  | FS.INI= 000000  | N\$MLL= 000001  | \$CMQRM 000047RG  |
| CP.PS= 177400  | C.MRKT= 000000 | FS.KIL= 000000  | N\$MOV= 000010  | \$CNV18 000114RG  |
| CP.PSI= 000200 | C.NSP= 000004  | FS.LCL= 100000  | N\$NCT= 000001  | \$CNV22 000114RG  |
| CP.XCF= 000100 | C.PRO= 000042  | FS.LTM= 001000  | N\$PEM= 000001  | \$DIV = ***** GX  |
| CP.ZFR= 000030 | C.RQT= 000002  | FS.MNT= 004000  | OFS = 000006    | \$MUL = ***** GX  |
| CS.ABO= 000100 | C.RSI= 000012  | FS.MSN= 014000  | P\$P45= 000000  | \$MVFBF 000242RG  |
| CS.BRO= 000002 | C.RSV= 000002  | FS.REA= 001000  | P\$WRD= 000000  | \$MVBFB 000176RG  |
| CS.BUF= 000200 | C.SCHD= 000002 | FS.RET= 000000  | Q\$OPT= 000010  | \$PDVID 000514RG  |
| CS.CES= 000002 | C.SRC= 000014  | FS.REZ= 003000  | R\$DER= 000000  | \$PDVNM= ***** GX |
| CS.CHN= 000010 | C.SSHT= 000004 | FS.RLB= 002000  | R\$K11= 000001  | \$PDVTA= ***** GX |
| CS.CMP= 000200 | C.STA= 000007  | FS.RNG= 011000  | R\$SND= 000000  | \$PUMR = ***** GX |
| CS.DCR= 000400 | C.STS= 000012  | FS.RST= 000000  | R\$11M= 000000  | \$XBIAS= ***** GX |
| CS.DEF= 000004 | C.SUB= 000012  | FS.RTN= 001000  | SS\$WRG= 000000 | .\$\$\$\$= 000034 |
| CS.DEV= 000002 | C.SYST= 000006 | FS.SET= 005000  |                 |                   |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)

```

183 .SBTTL $RDBRT - Return a Receive Data Buffer
184 .SBTTL $LDBRT - Return a Large Data Buffer
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209 000170 $LDBRT::
210 000170 $RDBRT:: SAVRG <R2,R3,R5> ; Save some registers
211 000176 INHIB$;;; Inhibit Interrupts
212
213 000210 016464 000026 000016 MOV C.BUF2+2(R4),C.BUF+2(R4)
214
215 000216 012705 000000G MOV #RDBCT,R5 ;;; Copy the buffer's Virtual Address
216 000222 005767 000000G TST $RDQCT ;;; Point to count in allocation control block
217 000226 003475 BLE RELBUF ;;; Are any processes waiting for an RDB?
218 000230 012703 100000 MOV #100000,R3 ;;; If LE, no waiters or buffer already queued
219 000234 050367 000000G BIS R3,$RDQCT ;;; Else, send to buffer return system process
220 000240 012764 001002 000010 MOV #FC.RCE!FS.RTN,C.FNC(R4) ;;; Mark a buffer return in progress
221
222
223
224
225
226 000246 005014 CLR (R4) ;;; Clear link word (only return 1 buffer)
227 000250 $PDQUE ;;; Queue buffer to system process
228 000254 000472 BR CMNRTN ;;; Enter common code

```

|                  |                   |                  |                  |                     |
|------------------|-------------------|------------------|------------------|---------------------|
| A\$\$CHK= 000000 | CS.DCR= 000400    | D\$\$YNM= 000000 | I\$\$RAR= 000000 | M\$\$CRB= 000124    |
| A\$\$CPS= 000000 | CS.DEF= 000004    | E\$\$XPR= 000000 | I\$\$RDN= 000000 | M\$\$CRX= 000000    |
| A\$\$PRI= 000000 | CS.DEV= 000002    | FAIL 000660R     | KISAR6= ***** GX | M\$\$FCS= 000000    |
| A\$\$TRP= 000000 | CS.DIS= 000040    | FC.CCP= 000020   | K\$\$CNT= 177546 | M\$\$MGE= 000000    |
| BUFGT 000610R    | CS.ENA= 000001    | FC.CTL= 000006   | K\$\$CSR= 177546 | M\$\$NET= 000000    |
| BUFRT 000670R    | CS.ENB= 000020    | FC.KCP= 000016   | K\$\$LDC= 000000 | M\$\$OVR= 000000    |
| CB.CCB= 000002   | CS.ERR= 100000    | FC.KIL= 000004   | K\$\$TPS= 000074 | N\$\$ACC= 000001    |
| CB.DDM= 000040   | CS.FTL= 001000    | FC.MAN= 000024   | LD\$LP= 000000   | N\$\$BUF= 000001    |
| CB.DLC= 000020   | CS.HCR= 000001    | FC.MLD= 000026   | LF.ACT= 100000   | N\$\$LDV= 000001    |
| CB.RDB= 000004   | CS.HFE= 002000    | FC.PCT= 000030   | LF.BRO= 000400   | N\$\$MCP= 000001    |
| CB.SDB= 000010   | CS.LST= 040000    | FC.PWR= 000022   | LF.BWT= 000007   | N\$\$ML= 000001     |
| CB.SLI= 000100   | CS.MTL= 004000    | FC.RCE= 000002   | LF.ENA= 002000   | N\$\$MOV= 000010    |
| CB.XLB= 000001   | CS.RNG= 000010    | FC.RCP= 000014   | LF.LPB= 001000   | N\$\$NCT= 000001    |
| CCBGT 000534R    | CS.ROV= 000004    | FC.TIM= 000010   | LF.MDC= 000100   | N\$\$PEM= 000001    |
| CCBRT 000572R    | CS.RSN= 010000    | FC.XCP= 000012   | LF.MFL= 004000   | PRIOFF= 000000      |
| CC.LLC= 000200   | CS.SHU= 000001    | FC.XME= 000000   | LF.MTP= 000020   | PR7 = ***** GX      |
| CE.ABO= 100362   | CS.SID= 000002    | FS.AST= 000000   | LF.PAC= 000200   | PS = ***** GX       |
| CE.DAO= 100346   | CS.STR= 000004    | FS.CIB= 002000   | LF.RDY= 040000   | P\$\$P45= 000000    |
| CE.DIS= 100366   | CS.SUC= 000001    | FS.CRA= 001000   | LF.REA= 010000   | P\$\$WRD= 000000    |
| CE.ERR= 100370   | CS.TMO= 020000    | FS.DIS= 013000   | LF.SER= 000040   | Q\$\$OPT= 000010    |
| CE.ILN= 100350   | CS.XUR= 000004    | FS.DVC= 001000   | LF.TIM= 000010   | RELBUR 000422R      |
| CE.LTO= 100356   | C\$\$CKP= 000000  | FS.ENB= 012000   | LF.LNL= 020000   | R\$\$DER= 000000    |
| CE.MOP= 100372   | C\$\$ORE= 000400  | FS.EXI= 001000   | LF.X2P= 000000   | R\$\$K11= 000001    |
| CE.NTE= 100361   | C\$\$SRSH= 177564 | FS.GET= 006000   | LN.CLO= 000000   | R\$\$SND= 000000    |
| CE.RTE= 100376   | C.ADD 000034      | FS.HLT= 000000   | LN.DUM= 000005   | R\$\$T1M= 000000    |
| CE.SRC= 100364   | C.BID 000003      | FS.INI= 000000   | LN.LOA= 000004   | SF.ACT= 000200      |
| CE.STP= 100352   | C.BUF 000014      | FS.KIL= 000000   | LN.LOO= 000003   | SF.ENA= 000100      |
| CE.TME= 100354   | C.BUF1 000014     | FS.LCL= 100000   | LN.OAU= 000003   | SF.LPB= 000004      |
| CE.TMO= 100374   | C.BUF2 000024     | FS.LTM= 001000   | LN.OFF= 000001   | SF.MFL= 000040      |
| CE.UNS= 100344   | C.CNT 000020      | FS.MNT= 004000   | LN.ON = 000000   | SF.PAC= 000020      |
| CF.CHN= 000001   | C.CNT1 000020     | FS.MSN= 014000   | LN.OOP= 000004   | SF.REA= 000010      |
| CF.EQM= 000004   | C.CNT2 000030     | FS.REA= 001000   | LN.OPE= 000001   | SF.SER= 000001      |
| CF.HDR= 000020   | C.FLG 000022      | FS.RET= 000000   | LN.REF= 000002   | SF.SVC= 000002      |
| CF.LB = 100000   | C.FLG1 000022     | FS.REZ= 003000   | LN.SER= 000002   | SF.UNL= 000040      |
| CF.LIN= 000002   | C.FLG2 000032     | FS.RLB= 002000   | LN.STA= 000017   | S\$\$WRG= 000000    |
| CF.SOM= 000010   | C.FNC 000010      | FS.RNG= 011000   | LN.SUB= 000360   | S\$\$YSZ= 007600    |
| CF.SYN= 000040   | C.LIN 000006      | FS.RST= 000000   | LN.TRI= 000006   | S.COST 000001       |
| CF.TRN= 000100   | C.LNK 000000      | FS.RTN= 001000   | L\$\$ASG= 000000 | S.FLG 000000        |
| CMNRTN 000442R   | C.MOD 000011      | FS.SET= 005000   | L\$\$DRV= 000000 | S.LEN 000004        |
| CM.CIR= 000002   | C.NSP 000004      | FS.SFC= 005000   | L\$\$P11= 000001 | S.NMST 000002       |
| CM.FMT= 100000   | C.PRO 000042      | FS.SFR= 006000   | L\$\$11R= 000000 | S.OWNR 000003       |
| CM.HRD= 000002   | C.RSV 000002      | FS.SFS= 004000   | L.COST 000015    | T\$\$KMG= 000000    |
| CM.LIN= 000000   | C.STA 000007      | FS.SPW= 040000   | L.CTL 000012     | T\$\$MIN= 000000    |
| CM.LOO= 000001   | C.STS 000012      | FS.STM= 000000   | L.CVA 177776     | V\$\$CTR= 001000    |
| CM.XLO= 000004   | C.URM 177776      | FS.STP= 002000   | L.DDM 000002     | X\$\$DBT= 000000    |
| CP.DCF= 000040   | C.XACP 000004     | FS.STR= 001000   | L.DDS 000004     | Y\$\$BAF= ***** GX  |
| CP.HDL= 000007   | C.XID 000035      | FS.TRM= 003000   | L.DLC 000003     | Y\$\$CBT= ***** GX  |
| CP.PS = 177400   | C.XLEN= 000044    | FS.WLB= 001000   | L.DLM 000006     | Y\$\$CBT 000000RG   |
| CP.PSI= 000200   | C.XPLI 000040     | FS.XKL= 002000   | L.DLS 000010     | Y\$\$CLH= ***** GX  |
| CP.XCF= 000100   | C.XPT 000034      | FS.XOF= 010000   | L.FLG 000000     | Y\$\$CBRT 000256RG  |
| CP.2FR= 000030   | C.XSVC 000042     | FS.XON= 007000   | L.KRBA 000016    | Y\$\$CBGT 000302RG  |
| CS.ABO= 000100   | C.XTC 000037      | FS.ZER= 002000   | L.LEN = 000022   | Y\$\$CBRT 000376RG  |
| CS.BRO= 000002   | C.X25 000036      | FS\$LV= 000001   | L.MPF 000022     | Y\$\$DBAF= ***** GX |
| CS.BUF= 000200   | D\$\$BUG= 177514  | G\$\$TPP= 000000 | L.NMST 000020    | Y\$\$DBG 000026RG   |
| CS.CES= 000002   | D\$\$ISK= 000000  | G\$\$TSS= 000000 | L.NSTA 000014    | Y\$\$DBRT 000170RG  |
| CS.CHN= 000010   | D\$\$L11= 000001  | G\$\$TTK= 000000 | L.OWNR 000021    | Y\$\$DQUE= ***** GX |
| CS.CMP= 000200   | D\$\$YNC= 000000  | G\$\$WRD= 000000 | L.UNT 000013     | Y\$\$DBCT= ***** GX |

```

188 .SBTTL $RDBRT - Return a Receive Data Buffer
189 .SBTTL $LDBRT - Return a Large Data Buffer
190
191 ;+
192 **-$RDBRT- Return a Receive Data Buffer
193 **-$LDBRT- Return a Large Data Buffer
194
195 This routine is called to return a data buffer to the large buffer
196 pool. If the waiters count is zero or a buffer is already queued to
197 the system process to satisfy a wait request, the buffer is returned
198 to the pool. Otherwise, the system return process is scheduled.
199
200 Inputs:
201 R4 contains the address of buffer's CCB
202 Note that C.BUF2+2 must contain the original Virtual
203 Address of the buffer, and C.BUF must contain the original
204 Bias of the buffer.
205
206 Outputs:
207 None.
208
209 $LDBRT::
210 $RDBRT::SAVRG <R2,R3,R5> ; Save some registers
211 INHIB$;;; Inhibit Interrupts
212
213 000210 016464 000026 000016 MOV C.BUF2+2(R4),C.BUF+2(R4)
214 ;;; Copy the buffer's Virtual Address
215 000216 012705 000000G MOV #RDBCT,R5 ;;; Point to count in allocation control block
216 000222 005767 000000G TST $RDQCT ;;; Are any processes waiting for an RDB?
217 000226 003475 BLE RELBUF ;;; If LE, no waiters or buffer already queued
218 000230 012703 100000 MOV #100000,R3 ;;; Else, send to buffer return system process
219 000234 050367 000000G BIS R3,$RDQ ;;; Mark a buffer return in progress
220 000240 012764 001002 000010 MOV #FC.RCE:FS.RTN,C.FNC(R4)
221
222 .IF DF M$SPRO
223 CLR C.URM(R4) ;;; Allow process to run on any processor
224 .ENDC
225
226 000246 005014 CLR (R4) ;;; Clear link word (only return 1 buffer)
227 000250 CALL $PD^UE ;;; Queue buffer to system process
228 000254 000472 BR CMNRTN ;;; Enter common code

```

|                  |                  |                  |                  |                   |
|------------------|------------------|------------------|------------------|-------------------|
| A\$\$CHK= 000000 | CS.DCR= 000400   | D\$\$YNM= 000000 | I\$\$RAR= 000000 | M\$\$CRB= 000124  |
| A\$\$CPS= 000000 | CS.DEF= 000004   | E\$\$XPR= 000000 | I\$\$RDN= 000000 | M\$\$CRX= 000000  |
| A\$\$PRI= 000000 | CS.DEV= 000002   | FAIL 001036R     | KISAR6= ***** GX | M\$\$FCS= 000000  |
| A\$\$TRP= 000000 | CS.DIS= 000040   | FC.CCP= 000020   | K\$\$CNT= 177546 | M\$\$MGE= 000000  |
| BUFGT 000766R    | CS.ENA= 000001   | FC.CTL= 000006   | K\$\$CSR= 177546 | M\$\$NET= 000000  |
| BUFR 001046R     | CS.ENB= 000020   | FC.KCP= 000016   | K\$\$LDC= 000000 | M\$\$OVR= 000000  |
| CB.CCB= 000002   | CS.ERR= 100000   | FC.KIL= 000004   | K\$\$TPS= 000074 | N\$\$ACC= 000001  |
| CB.DDM= 000040   | CS.FTL= 001000   | FC.MAN= 000024   | LD\$LP= 000000   | N\$\$BUI= 000001  |
| CB.DLC= 000020   | CS.HCR= 000001   | FC.MLD= 000026   | LF.ACT= 100000   | N\$\$LDV= 000001  |
| CB.RDB= 000004   | CS.HFE= 002000   | FC.PCT= 000030   | LF.BRO= 000400   | N\$\$MCP= 000001  |
| CB.SDB= 000010   | CS.LST= 040000   | FC.PWR= 000022   | LF.BWT= 000007   | N\$\$MLL= 000001  |
| CB.SLI= 000100   | CS.MTL= 004000   | FC.RCE= 000002   | LF.ENA= 002000   | N\$\$MOV= 000010  |
| CB.XLB= 000001   | CS.RNG= 000010   | FC.RCP= 000014   | LF.LPB= 001000   | N\$\$NCT= 000001  |
| CCBGT 000534R    | CS.ROV= 000004   | FC.TIM= 000010   | LF.MDC= 000100   | N\$\$OPT= 000001  |
| CCBRT 000656R    | CS.RSN= 010000   | FC.XCP= 000012   | LF.MFL= 004000   | N\$\$PEM= 000001  |
| CC.LLC= 000200   | CS.SHU= 000001   | FC.XME= 000000   | LF.MTP= 000020   | PRIOFF= 000000    |
| CE.ABO= 100362   | CS.SID= 000002   | FS.AST= 000000   | LF.PAC= 000200   | PR7 = ***** GX    |
| CE.DAO= 100346   | CS.STR= 000004   | FS.CIB= 002000   | LF.RDY= 040000   | PS = ***** GX     |
| CE.DIS= 100366   | CS.SUC= 000001   | FS.CRA= 001000   | LF.REA= 010000   | P\$\$P45= 000000  |
| CE.ERR= 100370   | CS.TMO= 020000   | FS.DIS= 013000   | LF.SER= 000040   | P\$\$WRD= 000000  |
| CE.ILN= 100350   | CS.XUR= 000004   | FS.DVC= 001000   | LF.TIM= 000010   | Q\$\$OPT= 000010  |
| CE.LTO= 100356   | C\$\$CKP= 000000 | FS.ENB= 012000   | LF.UNL= 020000   | REIBUF 000422R    |
| CE.MOP= 100372   | C\$\$ORE= 000400 | FS.EXI= 001000   | LF.X2P= 000000   | R\$\$DER= 000000  |
| CE.NTE= 100361   | C\$\$RSH= 177564 | FS.GET= 006000   | LN.CLO= 000000   | R\$\$K11= 000001  |
| CE.RTE= 100376   | C.ADD 000034     | FS.HLT= 000000   | LN.DUM= 000005   | R\$\$SND= 000000  |
| CE.SRC= 100364   | C.BID 000003     | FS.INI= 000000   | LN.LOA= 000004   | R\$\$11M= 000000  |
| CE.STP= 100352   | C.BUF 000014     | FS.KIL= 000000   | LN.LOO= 000003   | SF.ACT= 000200    |
| CE.TME= 100354   | C.BUF1 000014    | FS.LCL= 100000   | LN.OAU= 000003   | SF.ENA= 000100    |
| CE.TMO= 100374   | C.BUF2 000024    | FS.LTM= 001000   | LN.OFF= 000001   | SF.LPB= 000004    |
| CE.UNS= 100344   | C.CNT 000020     | FS.MNT= 004000   | LN.ON = 000000   | SF.MFL= 000040    |
| CF.CHM= 000001   | C.CNT1 000020    | FS.MSN= 014000   | LN.OOP= 000004   | SF.PAC= 000020    |
| CF.EOM= 000004   | C.CNT2 000030    | FS.REA= 001000   | LN.OPE= 000001   | SF.REA= 000010    |
| CF.HDR= 000020   | C.FLG 000022     | FS.RET= 000000   | LN.REF= 000002   | SF.SER= 000001    |
| CF.LB = 100000   | C.FLG1 000022    | FS.REZ= 003000   | LN.SER= 000002   | SF.SVC= 000002    |
| CF.LIN= 000002   | C.FLG2 000032    | FS.RLB= 002000   | LN.STA= 000017   | SF.UNL= 000040    |
| CF.SOM= 000010   | C.FNC 000010     | FS.RNG= 011000   | LN.SUB= 000360   | S\$\$WRG= 000000  |
| CF.SYN= 000040   | C.LIN 000006     | FS.RST= 000000   | LN.TRI= 000006   | S\$\$YSZ= 007600  |
| CF.TRN= 000100   | C.LNK 000000     | FS.RTN= 001000   | L\$\$ASG= 000000 | S.COST 000001     |
| CMNRTN 000442R   | C.MOD 000011     | FS.SET= 005000   | L\$\$DRV= 000000 | S.FLG 000000      |
| CM.CIR= 000002   | C.NSP 000004     | FS.SFC= 005000   | L\$\$P11= 000001 | S.LEN 000004      |
| CM.FMT= 100000   | C.PRO 000042     | FS.SFR= 006000   | L\$\$11R= 000000 | S.NMST 000002     |
| CM.HRD= 000002   | C.RSV 000002     | FS.SFS= 004000   | L.COST 000015    | S.OWNR 000003     |
| CM.LIN= 000000   | C.STA 000007     | FS.SPW= 040000   | L.CTL 000012     | T\$\$KMG= 000000  |
| CM.LOO= 000001   | C.STS 000012     | FS.STM= 000000   | L.CVA 177776     | T\$\$MIN= 000000  |
| CM.XLO= 000004   | C.URM 177776     | FS.STP= 002000   | L.DDM 000002     | V\$\$CTR= 001000  |
| CP.DCF= 000040   | C.XACP 000004    | FS.STR= 001000   | L.DDS 000004     | X\$\$DBT= 000000  |
| CP.HDL= 000007   | C.XID 000035     | FS.TRM= 003000   | L.DLC 000003     | \$ALOCB= ***** GX |
| CP.PS = 177400   | C.XLEN 000044    | FS.WLB= 001000   | L.DLM 000006     | \$CCBAF= ***** GX |
| CP.PSI= 000200   | C.XPLI 000040    | FS.XKL= 002000   | L.DLS 000010     | \$CCBAL= ***** GX |
| CP.XCF= 000100   | C.XPT 000034     | FS.XOF= 010000   | L.FLG 000000     | \$CCBCT= ***** GX |
| CP.2FR= 000030   | C.XSVC 000042    | FS.XON= 007000   | L.KRBA 000016    | \$CCBGT 000000RG  |
| CS.ABO= 000100   | C.XTC 000037     | FS.ZER= 002000   | L.LEN = 000022   | \$CCBLH= ***** GX |
| CS.BRO= 000002   | C.X25 000036     | F\$\$LVL= 000001 | L.MPF 000022     | \$CCBRT 000256RG  |
| CS.BUF= 000200   | D\$\$BUG= 177514 | G\$\$TPP= 000000 | L.NMST 000020    | \$CCBSZ= ***** GX |
| CS.CES= 000002   | D\$\$ISK= 000000 | G\$\$TSS= 000000 | L.NSTA 000014    | \$CEAVL= ***** GX |
| CS.CHN= 000010   | D\$\$L11= 000001 | G\$\$TTK= 000000 | L.OWNR 000021    | \$CSBGT 000302RG  |
| CS.CMP= 000200   | D\$\$YNC= 000000 | G\$\$WRD= 000000 | L.UNT 000013     | \$CSBRT 000376RG  |



```

196 .SBTTL DDCM1 - COMMON PROCESS FOR SPECIAL NON-CCB FUNCTIONS
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234

```

```

 .ENABL LSB
DDCM1: .IF DF K$$$DAS

 MOV R3,$DDFNC ; STORE FUNCTION CODE IN DATA AREA
 MOV #$DDFNC,R3 ; POINT R3 AT FUNCTION CODE

 .ENDC

 MOV R2,-(SP) ; SAVE R2
 MOV 2(SP),R2 ; GET LINE NUMBER
 BR 22$; JOIN COMMON CODE TO DISPATCH TO DDM

```

```

 **DDCM1 - COMMON PROCESS FOR SPECIAL NON-CCB FUNCTIONS
 INPUTS:
 R3 = ADDRESS OF FUNCTION CODE
 R4 = UNSPECIFIED CALLING PARAMETER TO DDM
 STACK CONTAINS:
 00(SP) = LINE NUMBER (ORIGINALLY IN R3)
 02(SP) = RETURN ADDRESS TO CALLING DLC

 OUTPUTS TO DDM:
 R3 = SUBFUNCTION CODE (WORD INDEX)
 R4 = SAME AS ON ENTRY
 R5 = LINE TABLE ADDRESS

 OUTPUTS ON RETURN TO CALLER:
 FUNCTION HAS BEEN PERFORMED
 R4 = UNSPECIFIED RETURN PARAMETER TO DLC FROM DDM

 REGISTERS ACCROSS CALL:
 R2,R3,R5 PRESERVED
 R4 MAY BE MODIFIED BY DDM

```

```

000140
000140 010246
000142 016602 000002
000146 000410

```

CEDDM CREATED BY MACRO ON 28-JUN-85 AT 18:18 PAGE 2 J 6  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL    | VALUE    | REFERENCES                     |
|-----------|----------|--------------------------------|
| LN.LOO    | = 000003 | #4-68                          |
| LN.OAU    | = 000003 | #4-68                          |
| LN.OFF    | = 000001 | #4-68                          |
| LN.ON     | = 000000 | #4-68                          |
| LN.OOP    | = 000004 | #4-68                          |
| LN.OPE    | = 000001 | #4-68                          |
| LN.REF    | = 000002 | #4-68                          |
| LN.SER    | = 000002 | #4-68                          |
| LN.STA    | = 000017 | #4-68                          |
| LN.SUB    | = 000360 | #4-68                          |
| LN.TRI    | = 000006 | #4-68                          |
| L\$S11    | = *****  | 11-406                         |
| L.COST    | 000015   | #4-68                          |
| L.CTL     | 000012   | #4-68                          |
| L.CVA     | 177776   | #4-68                          |
| L.DDM     | 000002   | #4-68 14-577                   |
| L.DDS     | 000004   | #4-68 14-576                   |
| L.DLC     | 000003   | #4-68                          |
| L.DLM     | 000006   | #4-68                          |
| L.DLS     | 000010   | #4-68                          |
| L.FLG     | 000000   | #4-68                          |
| L.KRBA    | 000016   | #4-68                          |
| L.LEN     | = 000022 | #4-68                          |
| L.MPF     | 000022   | #4-68                          |
| L.NMST    | 000020   | #4-68                          |
| L.NSTA    | 000014   | #4-68                          |
| L.OWNR    | 000021   | #4-68                          |
| L.UNT     | 000013   | #4-68                          |
| M\$S\$PRO | = *****  | 13-516                         |
| N\$S\$1LN | = *****  | 14-558                         |
| PR7       | = *****  | 11-404 14-564                  |
| PS        | = *****  | 11-396 *11-404 *11-414 *11-421 |
| R\$S\$11D | = *****  | 4-67                           |
| R\$S\$11M | = 000000 | 4-67                           |
| R\$S\$11S | = *****  | 4-67                           |
| SF.ACT    | = 000200 | #4-68                          |
| SF.ENA    | = 000100 | #4-68                          |
| SF.LPB    | = 000004 | #4-68                          |
| SF.MFL    | = 000040 | #4-68                          |
| SF.PAC    | = 000020 | #4-68                          |
| SF.REA    | = 000010 | #4-68                          |
| SF.SER    | = 000001 | #4-68                          |
| SF.SVC    | = 000002 | #4-68                          |
| SF.UNL    | = 000040 | #4-68                          |
| S.COST    | 000001   | #4-68                          |
| S.FLG     | 000000   | #4-68                          |
| S.LEN     | 000004   | #4-68                          |
| S.NMST    | 000002   | #4-68                          |
| S.OWNR    | 000003   | #4-68                          |
| X\$M\$CB  | = *****  | 4-67 4-3 9-272                 |
| X\$M\$DC  | = *****  | 4-2 4-3 6-116                  |
| X\$M\$NDM | = *****  | 4-1 4-3 6-116                  |

```

293
294
295
296 000170 032732 000100 BIT #LF.MDC,@(R2)+ ; DOES LINE NEED MODEM CONTROL?
297 000174 001435 BEQ 20$; NO - JOIN COMMON CODE TO DISPATCH TO DDM
298 000176 010546 MOV R5,-(SP) ; SAVE REGISTER
299 000200 010346 MOV R3,-(SP) ; SAVE ADDRESS OF FUNCTION CODE
300 000202 005002 CLR R2 ; GET AUXILIARY PROCESS PDX INDEX (ALWAYS 0)
301 000204 017705 000000G MOV @SPDVT,R5 ; GET ADDRESS OF AUXILIARY PROCESS' PDV
302 ; ; THAT WORKS SINCE IT'S PDV INDEX = 0
303 000210 016505 000016 MOV Z,DAT(R5),R5 ; GET ADDRESS OF DATA BASE DESCRIPTOR BLOCK
304 000214 CALL $PD DSP ; DISPATCH TO MODEM CONTROLLER
305 000220 012603 MOV (SP)+,R3 ; RESTORE R3
306
307 000222 116402 000006 MOV C,LIN(R4),R2 ; OBTAIN LINE NUMBER (NO SIGN EXTENSION)
308 000226 052764 100000 000006 BIS #100000,C,LIN(R4) ; INDICATE MODEM CONTROL REQUEST
309 000234 CALL DDMDSP ; DISPATCH TO THE DEVICE DRIVER (DDM)
310 000240 103420 BCS 25$; REQUEST WILL COMPLETE ASYNCHRONOUSLY
311 000242 CALL $CCBRT ; OTHERWISE RETURN CCB TO POOL
312 000246 005004 CLR R4 ; MODEM CONTROLLER WILL RETURN ASYNCHRONOUS COMPLETION
313 000250 000261 SEC ; SET C-BIT, LINE IS MODEM CONTROLLED
314 000252 000413 BR 25$; JOIN COMMON CODE
315
 .ENDC

```

CEDDMM      CREATED BY    MACRO    ON 28-JUN-85 AT 18:18      PAGE 4      J 8

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE       | REFERENCES          |
|---------|-------------|---------------------|
| \$DDXON | 000116 RG   | #8-191              |
| \$LLCSP | 000500 RG   | #14-512             |
| \$PDDSP | =        GX | 10-304      12-419  |
| \$PDQU1 | =        GX | 14-527              |
| \$PDVTA | =        GX | 10-301      12-399  |
| \$SLTMA | =        GX | 10-292      15-571  |
| \$STDDM | 000510 RG   | #15-558             |
| \$STDD1 | 000514 RG   | 12-394      #15-564 |

374

.DSABL LSB

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57.TITLE CEDLC  
.IDENT /V05.00/COPYRIGHT (C) 1978,1979,1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

## MODULE DESCRIPTION:

CEX DLC TO LLC INTERFACE ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

## IDENT HISTORY:

- 1.00 10-FEB-78  
VERSION 2.0 RELEASE
- 2.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/Rsx V1.0

; MACRO LIBRARY CALLS

|    |     |                                                 |
|----|-----|-------------------------------------------------|
| 5- | 65  | \$LLCRQ - LLC TO DLC REQUEST QUEUING SUBROUTINE |
| 6- | 128 | \$LLCRS - LLC TO LLC REQUEST QUEUING SUBROUTINE |
| 7- | 182 | PROCESS TO PROCESS DIRECT CALL INTERFACE        |

\*\*FILE\*\*ID\*\*CELOG

```

CCCCCCCC EEEEEEEEE LL 000000 GGGGGGGG
CCCCCCCC EEEEEEEEE LL 000000 GGGGGGGG
CC EE LL 00 00 GG
CC EE LL 00 00 GG
CC EE LL 00 00 GG
CC EE LL 00 00 GG
CC EEEEEEEE LL 00 00 GG
CC EEEEEEEE LL 00 00 GG
CC EE LL 00 00 GG
CC EE LL 00 00 GG
CC EE LL 00 00 GG
CC EE LL 00 00 GG
CC EE LL 00 00 GG
CCCCCCCC EEEEEEEEE LLLLLLLLL 000000 GGGGGG
CCCCCCCC EEEEEEEEE LLLLLLLLL 000000 GGGGGG

```

....  
....  
....  
....

```

LL SSSSSSSS TTTTTTTTT
LL SSSSSSSS TTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLL SSSSSSS TT
LLLLLLLL SSSSSSS TT

```



|                  |                  |                  |                  |                   |
|------------------|------------------|------------------|------------------|-------------------|
| A\$\$CHK= 000000 | I\$\$RDN= 000000 | N\$\$LDV= 000001 | X\$\$DBT= 000000 | Z\$.ASN= 100000   |
| A\$\$CPS= 000000 | K\$\$CNT= 177546 | N\$\$MCP= 000001 | ZF.COU= 001000   | Z\$.BSY= 140000   |
| A\$\$PRI= 000000 | K\$\$CSR= 177546 | N\$\$MML= 000001 | ZF.DDM= 000001   | Z\$.AVL 000014    |
| A\$\$TRP= 000000 | K\$\$LDC= 000000 | N\$\$MOV= 000010 | ZF.DIA= 004000   | Z\$.DAT 000016    |
| C\$\$CKP= 000000 | K\$\$TPS= 000074 | N\$\$NCT= 000001 | ZF.DLC= 000002   | Z\$.DSP 000000    |
| C\$\$ORE= 000400 | LD\$LP = 000000  | N\$\$PEN= 000001 | ZF.DVP= 100000   | Z\$.FLG 000010    |
| C\$\$RSH= 177564 | L\$\$ASG= 000000 | P\$\$P45= 000000 | ZF.INI= 040000   | Z\$.LEN = 000016  |
| D\$\$BUG= 177514 | L\$\$DRV= 000000 | P\$\$WRD= 000000 | ZF.KMX= 000020   | Z\$.LLN 000006    |
| D\$\$ISK= 000000 | L\$\$P11= 000001 | Q\$\$OPT= 000010 | ZF.LLC= 000004   | Z\$.MAP 000020    |
| D\$\$L11= 000001 | L\$\$11R= 000000 | R\$\$DER= 000000 | ZF.LMC= 000100   | Z\$.NAM 000004    |
| D\$\$YNC= 000000 | M\$\$CRB= 000124 | R\$\$K11= 000001 | ZF.MAN= 020000   | Z\$.PCB 000012    |
| D\$\$YNM= 000000 | M\$\$CRX= 000000 | R\$\$SND= 000000 | ZF.MFL= 000010   | Z\$.SCH 000007    |
| E\$\$XPR= 000000 | M\$\$FCS= 000000 | R\$\$11M= 000000 | ZF.MTM= 000400   | \$CELFN= ***** GX |
| F\$\$LVL= 000001 | M\$\$MGE= 000000 | S\$\$WRG= 000000 | ZF.MUX= 000040   | \$CELOG 000000RG  |
| G\$\$TTP= 000000 | M\$\$NET= 000000 | S\$\$YSZ= 007600 | ZF.PSE= 002000   | \$CMPDV= ***** GX |
| G\$\$TSS= 000000 | M\$\$OVR= 000000 | T\$\$KMG= 000000 | ZF.SLI= 010000   | \$LGDDB= ***** GX |
| G\$\$TTK= 000000 | N\$\$ACI= 000001 | T\$\$MIN= 000000 | ZF.TIM= 000200   | \$LGPDV= ***** GX |
| G\$\$WRD= 000000 | N\$\$BUF= 000001 | V\$\$CTR= 001000 | ZF.X3P= 000000   | \$PDDSP= ***** GX |
| I\$\$RAR= 000000 | N\$\$EVL= 000001 |                  |                  |                   |

. ABS. 000020 000 (RW,I,GBL,ABS,OVR)  
000046 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 10191 Words ( 40 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:07.21  
SY:CELOG1.V2,[130,134]CELOG1/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]V2,CELOG

```

283 .SBTTL $PDQUE - QUEUE A CCB (CHAIN) TO A LIST AND SCHEDULE PROCESS
284
285 ;+
286 ;**-$PDQUE-QUEUE A CCB CHAIN TO A LIST AND SCHEDULE PROCESS
287
288 Inputs:
289
290 R3 = PDV Index (high byte) and channel number (low byte) ($PDQUE ONLY)
291 R4 = Address of first CCB in a chain. The last CCB of the chain
292 has a zero next CCB pointer.
293
294 Outputs:
295
296 The PDV Index and channel number parameters are stored in the CCB
297 and the CCB (or chain) is linked to the operation queue for the
298 process defined by the PDV index and a schedule request for that
299 process is set.
300
301 Registers modified:
302
303 R3 and R4
304
305 ;-
306 000170 010364 000006 $PDQUE::MOV R3,C.LIN(R4) ; Store PDV index and channel number
307
308 000174 010546 $PDQU1::MOV R5,~(SP) ; Save callers R5
309 000176 010405 MOV R4,R5 ; Copy address of first CCB
310
311 000200 042764 040000 000012 10$: BIC #CS.LST,C.STS(R4) ; Clear last CCB marker
312 000206 010403 MOV R4,R3 ; Copy current CCB address
313 000210 011304 MOV (R3),R4 ; Get next CCB address
314 000212 001372 BNE 10$; If ne keep going
315 000214 052765 040000 000012 BIS #CS.LST,C.STS(R3) ; Mark last CCB of chain
316 000222 012704 000004G MOV #SCFRK+4,R4 ; Set address of queue listhead
317 000226 INHIB$; Inhibit interrupts
318
319 .IF DF M$$PRO
320 CALL $MPLCK ;;; Lock access to process queue
321 .ENDC
322
323 000240 010574 000002 MOV R5,@2(R4) ;;; Link first to last
324 000244 010364 000002 MOV R3,2(R4) ;;; Set new last pointer
325 000250 024444 CMP ~(R4),~(R4) ;;; Fork block already in use ?
326 000252 001010 BNE 20$;;; If NE, yes - just exit
327 000254 005767 000000G TST $CXOPT ;;; Are we in fork processing?
328 000260 100405 BMI 20$;;; If MI, yes - just exit
329 000262 005014 CLR (R4) ;;; Clear next link pointer
330
331 .IF DF R$$MPL
332 .IF DF M$$PRO
333 MOV ~(R5),~2(R4) ;;; Set up required Unibus Run Mask
334 .ENDC ; DF M$$PRO
335
336 CALL $QFORK ;;; Queue the fork block
337
338 .IFF ; DF R$$MPL
339 000264 010477 000002G MOV R4,@$FRKHD+2 ;;; Link to end of fork list

```

```

101 .SBTTL $CMQRM - REMOVE A CHAIN OF CCBS FROM A LIST
102
103 ;+
104 ;**-$CMQRM - REMOVE A CHAIN OF CCBS FROM A LIST
105 ;
106 ; THIS SUBROUTINE IS CALLED TO REMOVE THE FIRST ENTRY
107 ; FROM A SINGLY LINKED LIST.
108
109 ; INPUTS:
110 ;
111 ; R3 = ADDRESS OF TWO WORD LISTHEAD
112
113 ; OUTPUTS:
114 ;
115 ; C/CLEAR:
116 ; R4 = ADDRESS OF FIRST IN CHAIN OR ONLY CCB DEQUEUED
117
118 ; C/SET: QUEUE IS EMPTY
119
120 ; REGISTERS MODIFIED:
121 ;
122 ; R4
123 ; -
124
125 000044 000261 $CMQRM::SEC ; SET C-BIT JUST IN CASE QUEUE IS EMPTY
126 000046 011304 MOV (R3),R4 ; GET ADDRESS OF FIRST CCB IN QUEUE
127 000050 001420 BEQ 40$; RETURN IF QUEUE IS EMPTY
128 000052 SAVRG <R5,R4> ; SAVE A REGISTER
129 000056 010405 10$: MOV R4,R5 ; COPY FIRST CCB IN CHAIN ADDRESS
130 000060 032765 040000 000012 BIT #CS.LST,C.STS(R5) ; IS THIS THE LAST CCB IN THE CHAIN?
131 000066 001002 BNE 20$; YES - UPDATE THE LISTHEAD
132 000070 011504 MOV (R5),R4 ; NO - GET NEXT CCB - IS THIS THE LAST?
133 000072 001371 BNE 10$; NO - LOOP TILL END OF CHAIN
134
135 000074 011513 20$: MOV (R5),(R3) ; SET ADDRESS OF NEW FIRST CCB IN QUEUE
136 000076 001002 BNE 30$; BRANCH IF QUEUE IS NOT EMPTY
137 000100 010363 000002 MOV R3,2(R3) ; OTHERWISE CLOSE UP LIST
138 000104 005015 30$: CLR (R5) ; CLEAR LINK POINTER IN THE LAST CCB OF CHAIN
139 000106 RESRG <R4,R5> ; RESTORE REGISTER
140 000112 40$: RETURN ; RETURN

```

CESUB MACRO V05.03b Friday 28-Jun-85 18:20 Page 16-2  
Symbol table

J 16

000602 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 15909 Words ( 63 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:16.49

SY:CESUB.V2,[130,134]CESUB/CR/--SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]CESUB

```

230 .SBTTL $CCBRT - Return a CCB
231 ;+
232 **-$CCBRT- Return a CCB
233 This routine is called to deallocate a standard CCB.
234 Inputs:
235 R4 contains the address of the CCB
236 Outputs:
237 None.
238 ;+
239 ;+
240 ;+
241 ;+
242 ;+
243 000256 $CCBRT::INHIB$;;; Inhibit Interrupts
244
245 .IF DF M$$PRO
246 CALL $MPLCK ;;; Lock access to CommExec resources
247 .ENDC
248
249 000270 CALL CCBRT ;;; Return the CCB
250
251 .IF DF M$$PRO
252 CALL @($P)+ ;;; Co-Routine return to unlock resources
253 .ENDC
254
255 000274 ENABL$; Enable Interrupts
256 000300 RETURN

```

CEBUF - CEX BUFFER MANAGEMENT R MACRO V05.03b Friday 28-Jun-85 18:16<sup>K 2</sup> Page 20-2  
Symbol table

|          |          |          |          |          |          |          |          |            |          |
|----------|----------|----------|----------|----------|----------|----------|----------|------------|----------|
| \$RDBG   | C00044RG | \$RDBRT  | 000170RG | \$RDBWT  | 000456RG | \$SDBLH= | ***** GX | \$SLTMA=   | ***** GX |
| \$RDBLH= | ***** GX | \$RDBSZ= | ***** GX | \$RDQCT= | ***** GX | \$SDBSZ= | ***** GX | .\$\$\$\$= | 000034   |
| \$RDBOP  | 000500RG | \$RDBTH= | ***** GX | \$SDBCT= | ***** GX |          |          |            |          |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000722 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 16471 Words ( 65 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:20.11  
SY:CEBUF,V2,[130,134]CEBUF/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]CEBUF

```

230 .SBTTL $CCBRT - Return a CCB
231 :+
232 **-$CCBRT- Return a CCB
233 :
234 This routine is called to deallocate a standard CCB.
235 :
236 Inputs:
237 R4 contains the address of the CCB
238 :
239 Outputs:
240 None.
241 :-
242
243 000256 $CCBRT::INHIB$::: Inhibit Interrupts
244
245 .IF DF M$$PRO
246 CALL $MPLCK ::: Lock access to CommExec resources
247 .ENDC
248
249 000270 CALL CCBRT ::: Return the CCB
250
251 .IF DF M$$PRO
252 CALL @($P)+ ::: Co-Routine return to unlock resources
253 .ENDC
254
255 000274 ENAB$; Enable Interrupts
256 000300 RETURN

```

CEBUF1 - CEX BUFFER MANAGEMENT MACRO V05.03b Friday 28-Jun-85 18:17<sup>K 4</sup> Page 21-2  
Symbol table

|                   |                   |                   |                   |                   |
|-------------------|-------------------|-------------------|-------------------|-------------------|
| \$DEACB= ***** GX | \$PDQUE= ***** GX | \$RDBNM= ***** GX | \$RDBTH= ***** GX | \$SDBLH= ***** GX |
| \$LDBAF= ***** GX | \$RDBCT= ***** GX | \$RDBOP 000500RG  | \$RDBWT 000456RG  | \$SDBSZ= ***** GX |
| \$LDBG1 000026RG  | \$RDBG1 000044RG  | \$RDBRT 000170RG  | \$RDBCT= ***** GX | \$SLTMA= ***** GX |
| \$LDBRT 000170RG  | \$RDBLH= ***** GX | \$RDBSZ= ***** GX | \$SDBCT= ***** GX | .\$\$\$\$= 000034 |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
001100 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 16471 Words ( 65 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:21.13  
SY:CEBUF1.V2,[130,134]CEBUF1/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]V2,CEBUF



236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292

000150

```
.SBTTL DDCM2 - COMMON PROCESS FOR MODEM CONTROL FUNCTIONS

*
**--DDCM2 - COMMON PROCESS FOR MODEM CONTROL FUNCTIONS
IF THE SPECIFIED LINE HAS MODEM CONTROL ENABLED THE
FUNCTION REQUEST IS DERAILED TO THE MODEM CONTROLLER.
ON RETURN FROM THE MODEM CONTROLLER THE REQUEST IS
ROUTED TO THE PROPER DDM AS USUAL. IF THE LINE IS
HARDWIRED THE REQUEST IS ROUTED DIRECTLY TO THE DDM.
INPUTS:
R3 = ADDRESS OF FUNCTION CODE
R4 = ADDRESS OF CCB
STACK CONTAINS:
00(SP) = CALLER'S R3
02(SP) = RETURN ADDRESS TO CALLING DLC

OUTPUTS TO MODEM CONTROLLER:
R3 = SUBFUNCTION CODE
R5 = LINE TABLE ADDRESS

OUTPUTS TO DDM:
R3 = SUBFUNCTION CODE
R4 = SAME AS ON ENTRY
R5 = LINE TABLE ADDRESS

OUTPUTS ON RETURN TO CALLER:
C-BIT CLEAR - FUNCTION PERFORMED SYNCHRONOUSLY
C-BIT SET - WAIT FOR ASYNCHRONOUS REPORT OF FUNCTION COMPLETION

REGISTERS ACROSS CALL:
R2,R3,R5 - PRESERVED
R4 - MODIFIED
-

DDCM2:
 .IF DF X$$MDC
 .IF DF K$$DAS
 MOV R3,$DDFNC ; STORE THE FUNCTION CODE IN DATA AREA
 MOV #DDFNC,R3 ; POINT R3 AT FUNCTION CODE
 .ENDC
 MOV (R3),C.FNC(R4) ; PUT FUNCTION CODE IN CCB
 MOV R2,-(SP) ; SAVE R2
 .IF DF N$$1LN
 MOV $SLTMA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE
 .IFF
 MOVB C.LIN(R4),R2 ; GET SYSTEM LINE NUMBER
 ASL R2 ; FORM WORD INDEX
 ADD $SLTMA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE
```

CEDDM      CREATED BY MACRO ON 28-JUN-85 AT 18:18      PAGE 3      K 6  
 SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL  | VALUE      | REFERENCES        |
|---------|------------|-------------------|
| ZF.COU  | = 001000   | #4-67             |
| ZF.DDM  | = 000001   | #4-67             |
| ZF.DIA  | = 004000   | #4-67             |
| ZF.DLC  | = 000002   | #4-67             |
| ZF.DVP  | = 100000   | #4-67      11-401 |
| ZF.INI  | = 040000   | #4-67             |
| ZF.KMX  | = 000020   | #4-67             |
| ZF.LLC  | = 000004   | #4-67             |
| ZF.LMC  | = 000100   | #4-67             |
| ZF.MAN  | = 020000   | #4-67             |
| ZF.MFL  | = 000010   | #4-67             |
| ZF.MTM  | = 000400   | #4-67             |
| ZF.MUX  | = 000040   | #4-67             |
| ZF.PSE  | = 002000   | #4-67             |
| ZF.SLI  | = 010000   | #4-67             |
| ZF.TIM  | = 000200   | #4-67             |
| ZF.X3P  | = 000000   | #4-67             |
| ZS.ASN  | = 100000   | #4-67             |
| ZS.BSY  | = 140000   | #4-67             |
| Z.AVL   | 000014     | #4-67             |
| Z.DAT   | 000016     | #4-67             |
| Z.DSP   | 000000     | #4-67      4-67   |
| Z.FLG   | 000010     | #4-67      11-401 |
| Z.LEN   | = 000016   | #4-67             |
| Z.LLN   | 000006     | #4-67             |
| Z.MAP   | 000020     | #4-67             |
| Z.NAM   | 000004     | #4-67             |
| Z.PCB   | 000012     | #4-67             |
| Z.SCH   | 000007     | #4-67      11-414 |
| \$CCBGT | = ***** GX | 12-453            |
| \$CCBRT | = ***** GX | 13-508            |
| \$DDAST | 000274 RG  | #12-452           |
| \$DDCCP | 000354 RG  | #13-506           |
| \$DDCRA | 000030 RG  | #7-180            |
| \$DDDIS | 000110 RG  | #7-189            |
| \$DDENB | 000102 RG  | #7-188            |
| \$DDGET | 000060 RG  | #7-184            |
| \$DDKCP | 000344 RG  | #13-503           |
| \$DDKIL | 000036 RG  | #7-181            |
| \$DDMAN | 000014 RG  | #7-177            |
| \$DDMSN | 000132 RG  | #7-193            |
| \$DDRCE | 000006 RG  | #7-175            |
| \$DDRCR | 000334 RG  | #13-500           |
| \$DDRNG | 000074 RG  | #7-187            |
| \$DDSET | 000066 RG  | #7-185            |
| \$DDSPC | 000374 RG  | #13-513           |
| \$DDSTP | 000052 RG  | #7-183            |
| \$DDSTR | 000044 RG  | #7-182            |
| \$DDXKL | 000022 RG  | #7-179            |
| \$DDXME | 000000 RG  | #7-174            |
| \$DDXMP | 000324 RG  | #13-497           |
| \$DDXOF | 000124 RG  | #7-192            |

```

317 .SBTTL DDCM3 - COMMON PROCESS FOR FUNCTIONS WITH CCB
318 .SBTTL DDCM4 - COMMON PROCESS FOR FUNCTIONS WITH CCB
319
320 ;+
321 **DDCM3 - COMMON PROCESS FOR FUNCTIONS WITH CCB
322 **DDCM4 - COMMON PROCESS FOR FUNCTIONS WITH CCB
323
324 INPUTS:
325 R3 = ADDRESS OF FUNCTION AND SUBFUNCTION CODE (DDCM3)
326 R3 = ADDRESS OF FUNCTION CODE ONLY (DDCM4)
327 R4 = ADDRESS OF CCB
328 STACK CONTAINS:
329 00(SP) = CALLERS R3
330 02(SP) = RETURN ADDRESS TO CALLING DLC
331
332 OUTPUTS TO DDM:
333 R3 = SUBFUNCTION CODE (WORD INDEX)
334 R4 = ADDRESS OF CCB
335 R5 = ADDRESS OF LINE TABLE
336
337 REGISTERS ACROSS CALL:
338 R2,R3,R5 - PRESERVED
339 R4 - MAY BE MODIFIED
340
341
342 000254 DDCM3:
343 .IF DF K$$DAS
344
345 MOV R3,$DDFNC ; STORE FUNCTION CODE IN DATA AREA
346 MOV #DDFNC,R3 ; POINT R3 AT FUNCTION CODE
347
348 .IFTF ; DF K$$DAS
349
350 000254 011364 000010 MOV (R3),C.FNC(R4) ; PUT FUNCTION AND SUBFUNCTION IN CCB
351 000260 000402 BR 10$; JOIN COMMON CODE
352
353 000262 DDCM4:
354 .IFT ; DF K$$DAS
355
356 MOV R3,$DDFNC ; STORE FUNCTION CODE IN DATA AREA
357 MOV #DDFNC,R3 ; POINT R3 AT FUNCTION CODE
358
359 .ENDC ; DF K$$DAS
360
361 000262 111364 000010 MOV (R3),C.FNC(R4) ; PUT FUNCTION CODE IN CCB
362
363 000266 010246 10$: MOV R2,-(SP) ; SAVE R2
364
365 000270 116402 000006 20$: MOV C.LIN(R4),R2 ; GET SLN
366
367 000274 010546 22$: MOV R5,-(SP) ; SAVE R5
368 000276 CALL DDMDSP ; DISPATCH TO THE DEVICE DRIVER (DDM)
369 000302 012605 25$: MOV (SP)+,R5 ; RESTORE R5
370 000304 012602 MOV (SP)+,R2 ; RESTORE R2
371 000306 012603 MOV (SP)+,R3 ; RESTORE R3
372 000310 RETURN ; RETURN TO CALLING DLC
373

```

CEDDM    CREATED BY    MACRO    ON 28-JUN-85 AT 18:18    PAGE 5    K 8  
 MACRO CROSS REFERENCE    CREF    04.00

| MACRO NAME | REFERENCES                                                                             |
|------------|----------------------------------------------------------------------------------------|
| CALL       | 10-304    10-309    10-311    11-368    12-394    12-419    13-453                     |
| CALLR      | #5-65    14-508    14-527                                                              |
| CCBDF\$    | #5-64    5-66                                                                          |
| DDFDF      | #6-97    8-174    8-175    8-177    8-179    8-180    8-181    8-182    8-183    8-184 |
|            | 8-185    8-187    8-188    8-189    8-191    8-192    8-193                            |
| ENABL\$    | #5-63                                                                                  |
| INHIB\$    | #5-63                                                                                  |
| MFPS       | 12-396                                                                                 |
| MTPS       | 12-404    12-421                                                                       |
| PDVDF\$    | #5-64    5-67                                                                          |
| RESRG      | #5-63    12-418                                                                        |
| RETURN     | 11-372    12-422    13-460    15-578                                                   |
| SAVRG      | #5-63    12-397                                                                        |
| SLTDF\$    | #5-64    5-68                                                                          |

```

376 .SBTTL DDMDSP - DISPATCH TO DDM LEVEL
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422

```

```

 .SBTTL DDMDSP - DISPATCH TO DDM LEVEL
 **--DDMDSP-DISPATCH TO DDM LEVEL
 THIS SUBROUTINE IS CALLED TO DISPATCH DOWN TO THE DDM LEVEL. PROCESS
 DISPATCH WILL OCCUR AT THE CURRENT PRIORITY OR AT THE PROCESS PRIORITY
 DEPENDENT ON A PDV FLAG.
 INPUTS:
 R2 = SYSTEM LINE NUMBER
 REGISTERS MODIFIED:
 R5
 DDMDSP: CALL $STDD1 ; SET UP DDM PDV INDEX & LINE TABLE ADDRESS
 CLC ; SAVE CURRENT PRIORITY
 MFPS -(SP) ; WITH C-BIT CLEAR
 SAVRG <R0> ; GET A FREE REGISTER
 MOV R2,R0 ; COPY DDM PDV INDEX
 ADD $PDVTA,R0 ; POINT INTO PDV INDEX TABLE
 MOV (R0),R0 ; GET POINTER TO PDV
 BIT #ZF.DVP,Z.FLG(R0)
 BEQ 10$; IF EQ, RUN DDM AT CURRENT PRIORITY
 MTPS #PR7 ; RAISE PROCESSOR PRIORITY
 .IF DF L$$S11
 MFPS -(SP) ;;; GET CURRENT PRIORITY
 BICB Z.SCH(R0),(SP) ;;; SET UP PROCESS PRIORITY
 MTPS (SP)+ ;;; SET UP PROCESSOR PRIORITY
 .IFF
 BICB Z.SCH(R0),PS ;;; DROP TO PROCESS PRIORITY
 .ENDC
 10$: RESRG <R0> ; RESTORE REGISTER
 CALL $PDDSP ; DISPATCH TO THE DDM
 ADC (SP) ; UPDATE C-BIT IN SAVED PS
 MTPS (SP)+ ; RESTORE PROCESSOR PRIORITY
 RETURN

```

```
58
59
60
61
62 000000
63 000000
64 000000
65
;
 .MCALL SAVRG,RESRG
 .MCALL CCBDF$,PDVDF$,SLTDF$
 .MCALL CALLR
 CCBDF$; AVOID SYSTEM DEPENDENCY
 PDVDF$; DEFINE CCB OFFSETS
 SLTDF$; DEFINE PDV OFFSETS
 ; DEFINE SLT OFFSETS
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

```
.TITLE CELLC
.IDENT /V05.00/

: COPYRIGHT (C) 1978,1979,1980, 1982, 1983, 1985 BY
: DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

: THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
: ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
: INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
: COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
: OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
: TRANSFERRED.

: THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
: AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
: CORPORATION.

: DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
: SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

: MODULE DESCRIPTION:

: CEX LLC TO DLC & LLC INTERFACE ROUTINES

: DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

: IDENT HISTORY:

: 1.00 10-FEB-78
: VERSION 2.0 RELEASE

: 2.00 14-DEC-79
: DECNET-11M/S V3.0
: DECNET-11M-PLUS V1.0

: 3.00 16-APR-82
: DECNET-11M V3.1
: DECNET-11M-PLUS V1.1

: 4.00 07-NOV-83
: DECNET-11M V4.0
: DECNET-11M-PLUS V2.0

: 5.00 22-JUL-85
: DECnet-11M/S V4.2
: DECnet-11M-Plus V3.0
: DECnet-Micro/RSX V1.0

:
:
: MACRO LIBRARY CALLS
:
```

CELOG - COMM/EXEC EVENT LOGGING MACRO V05.03b Friday 28-Jun-85 18:19<sup>K 12</sup>  
Table of contents

5- 56 LOG NETWORK EVENT



CELOG1      CREATED BY    MACRO    ON 28-JUN-85 AT 18:20      PAGE 1      K 13

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE          | REFERENCES                        |
|---------|----------------|-----------------------------------|
| ISSAS   | = *****        | 5-54                              |
| NSSEVL  | = 000001       | #4-2      5-1      5-2      6-104 |
| RSS11D  | = *****        | 5-54                              |
| RSS11M  | = 000000       | 5-54                              |
| RSS11S  | = *****        | 5-54                              |
| XSEM CB | = *****        | 5-54      5-54                    |
| ZF.COU  | = 001000       | #5-54                             |
| ZF.DDM  | = 000001       | #5-54                             |
| ZF.DIA  | = 004000       | #5-54                             |
| ZF.DLC  | = 000002       | #5-54                             |
| ZF.DVP  | = 100000       | #5-54                             |
| ZF.INI  | = 040000       | #5-54                             |
| ZF.KMX  | = 000020       | #5-54                             |
| ZF.LLC  | = 000004       | #5-54                             |
| ZF.LMC  | = 000100       | #5-54                             |
| ZF.MAN  | = 020000       | #5-54                             |
| ZF.MFL  | = 000010       | #5-54                             |
| ZF.MTM  | = 000400       | #5-54                             |
| ZF.MUX  | = 000040       | #5-54                             |
| ZF.PSE  | = 002000       | #5-54                             |
| ZF.SLI  | = 010000       | #5-54                             |
| ZF.TIM  | = 000200       | #5-54                             |
| ZF.X3P  | = 000000       | #5-54                             |
| ZS.ASN  | = 100000       | #5-54                             |
| ZS.BSY  | = 140000       | #5-54                             |
| Z.AVL   | 000014         | #5-54                             |
| Z.DAT   | 000016         | #5-54                             |
| Z.DSP   | 000000         | #5-54      5-54                   |
| Z.FLG   | 000010         | #5-54                             |
| Z.LEN   | = 000016       | #5-54                             |
| Z.LLN   | 000006         | #5-54                             |
| Z.MAP   | 000020         | #5-54                             |
| Z.NAM   | 000004         | #5-54                             |
| Z.PCB   | 000012         | #5-54                             |
| Z.SCH   | 000007         | #5-54                             |
| \$CELFN | = *****    GX  | 6-116                             |
| \$CELOG | = 000000    RG | #6-104                            |
| \$CMPDV | = *****    GX  | 6-109                             |
| \$LGDDB | = *****    GX  | 6-115                             |
| \$LGPDV | = *****    GX  | 6-106                             |
| \$PDDSP | = *****    GX  | 6-117                             |

CESCH MACRO V05.03b Friday 28-Jun-85 18:20 Page 9-1  
\$PDQUE - QUEUE A CCB (CHAIN) TO A LIST AND SCHEDULE PROCESS

K 14

```
340 000270 010467 000002G MOV R4,$FRKHD+2 ::: Set new last pointer
341 .ENDC ; DF R$$MPL.
342
343 000274 20$: .IF DF M$$PRO
344 CALL @ (SP)+ ::: Co-routine return to unlock process queue
345 .ENDC
346
347 ENABL$::: Enable Interrupts
348 000274 MOV (SP)+,R5 ; Recover R5
349 000300 012605 RETURN ; Return to caller
350 000302
```

```

142 .SBTTL $CNV18 - CONVERT TO 18-BIT UNIBUS ADDRESS
143
144 ***$CNV18-CONVERT TO 18-BIT UNIBUS ADDRESS
145 ***$CNV22-CONVERT TO 22-BIT QBUS-22 ADDRESS
146
147 THIS SUBROUTINE IS CALLED TO CONVERT AN ADDRESS DOUBLEWORD
148 TO AN 18-BIT UNIBUS VIRTUAL ADDRESS. NOTE THAT THE SUBROUTINE
149 TO CONVERT TO A QBUS-22 ADDRESS IS THE SAME (AT THE MOMENT).
150
151 INPUTS:
152
153 R2 - VIRTUAL ADDRESS
154 R3 - RELOCATION BIAS (PHYSICAL ADDRESS/100)
155
156 OUTPUTS:
157
158 R2 - LOW ORDER 16 BITS OF UNIBUS ADDRESS
159 R3 - BITS 16 & 17 OF UNIBUS ADDRESS IN BITS 0 & 1
160
161
162
163 000114 .ENABL LSB ; Reference label
164 $CNV22::
165 .IF DF,K$$DAS ; Save virtual address
166 MOV R2,-(SP) ; Buffer from ex.c. or common pool?
167 TST R3 ; If NE, from common pool - use bias supplied
168 BNE 5$; Get virtual in required register
169 MOV R2,R3 ; Set up for left shift
170 CLR R2 ; Shift active page field from virtual
171 ASHC #3,R2 ; Make it a word index
172 ASL R2 ; Calculate D-space APR address
173 ADD #KDSARQ,R2 ; Get contents of APR
174 MOV (R2),R3 ; Join common code
175 BR 5$
176 .ENDC ; DF,K$$DAS
177 000114 $CNV18:: ; Reference label
178 000114 TST R3 ; Buffer from exec. or comm pool?
179 000116 BEQ 10$; If FQ, it's from exec
180 ; (in the low 16 or 20 K)
181 000120 ADD $PUMR,R3 ; Exec phys addr = virtual addr = unibus addr)
182 000124 MOV R2,-(SP) ; Add starting UMR bias to block address
183 000126 ; Save virtual
184 000126 5$: BIC #160000,(SP) ; Clear APR selector from virtual address
185 000132 CLR R2 ; Initialize 18 bit value
186 .IF DF R$$EIS ; ALIGN AS AN 18 BIT QUANTITY
187 ASHC #6,R2
188 .IFF ; DF R$$EIS
189
190 000006 .REPT 6
191 ASL R3 ; SHIFT LOW ORDER BITS
192 .NDR ; INTO HIGH ORDER 2 BITS FOR 18-BIT VALUE
193
194 .ENDC ; DF R$$EIS
195
196 000164 ADD R3,(SP) ; ADD LOW 16 BITS TO SAVED OFFSET IN BLOCK
197
198

```

CESUB      CREATED BY    MACRO    ON 28-JUN-85 AT 18:20      PAGE 1      K 16  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE          | REFERENCES                                                                                 |
|---------|----------------|--------------------------------------------------------------------------------------------|
| CS.LST  | = 040000       | 5-92      5-95      6-130                                                                  |
| C.STS   | = 000012       | *5-92      *5-95      6-130                                                                |
| IS\$AS  | = *****        | 4-64                                                                                       |
| KISAR5  | = *****    GX  | 8-245      *8-246      9-294      *9-295      *9-303      10-327      *10-341      *10-361 |
| KISAR6  | = *****    GX  | *11-390      *11-411                                                                       |
| K\$SDAS | = *****        | 7-164      10-323      10-324      10-329      10-343      10-357                          |
| M\$SPRO | = *****        | 15-515                                                                                     |
| OF.S    | = 000006       | #10-323      10-350      *10-351                                                           |
| R\$EIS  | = *****        | 4-1      4-2      7-186      11-398      13-470      14-501                                |
| R\$11D  | = *****        | 4-64                                                                                       |
| R\$11M  | = 000000       | 4-64                                                                                       |
| R\$11S  | = *****        | 4-64                                                                                       |
| X\$MCB  | = *****        | 4-64      4-64                                                                             |
| ZF.COU  | = 001000       | #4-64                                                                                      |
| ZF.DDM  | = 000001       | #4-64                                                                                      |
| ZF.DJA  | = 004000       | #4-64                                                                                      |
| ZF.DLC  | = 000002       | #4-64                                                                                      |
| ZF.DVP  | = 100000       | #4-64                                                                                      |
| ZF.INI  | = 040000       | #4-64                                                                                      |
| ZF.KMX  | = 000020       | #4-64                                                                                      |
| ZF.LLC  | = 000004       | #4-64                                                                                      |
| ZF.LMC  | = 000100       | #4-64                                                                                      |
| ZF.MAN  | = 020000       | #4-64                                                                                      |
| ZF.MFL  | = 000010       | #4-64                                                                                      |
| ZF.MTM  | = 000400       | #4-64                                                                                      |
| ZF.MUX  | = 000040       | #4-64                                                                                      |
| ZF.PSE  | = 002000       | #4-64                                                                                      |
| ZF.SLI  | = 010000       | #4-64                                                                                      |
| ZF.TIM  | = 000200       | #4-64                                                                                      |
| ZF.X3P  | = 000000       | #4-64                                                                                      |
| ZS.ASN  | = 100000       | #4-64                                                                                      |
| ZS.BSY  | = 140000       | #4-64                                                                                      |
| Z.AVL   | = 000014       | #4-64                                                                                      |
| Z.DAT   | = 000016       | #4-64                                                                                      |
| Z.DSP   | = 000000       | #4-64      4-64                                                                            |
| Z.FLG   | = 000010       | #4-64                                                                                      |
| Z.LEN   | = 000016       | #4-64                                                                                      |
| Z.LLN   | = 000006       | #4-64                                                                                      |
| Z.MAP   | = 000020       | #4-64                                                                                      |
| Z.NAM   | = 000004       | #4-64      10-338      12-435                                                              |
| Z.PCB   | = 000012       | #4-64                                                                                      |
| Z.SCH   | = 000007       | #4-64                                                                                      |
| \$CALLX | = 000316    RG | #10-326                                                                                    |
| \$CEACC | = 000424    RG | #11-390                                                                                    |
| \$CECAC | = 000432    RG | #11-391                                                                                    |
| \$CEDIV | = 000576    RG | #14-501                                                                                    |
| \$CEMUL | = 000572    RG | #13-470                                                                                    |
| \$CMPDV | = *****    GX  | 10-326      *10-349      *10-362                                                           |
| \$CMQIN | = 000000    RG | #5-90                                                                                      |
| \$LMGRM | = 000044    RG | #6-125                                                                                     |
| \$CNV18 | = 000114    RG | #7-176                                                                                     |
| \$CNV22 | = 000114    RG | #7-163                                                                                     |

```

258 .SBTTL $CSBGT - Get a CCB and a Small Data Buffer
259
260 **-$CSBGT- Get a CCB and a Small Data Buffer
261
262 This routine is called to allocate a CCB and a Small Data Buffer.
263 The routine also sets up the buffer descriptor in the CCB to point
264 to the buffer as well as the proper byte count.
265
266 Inputs:
267 None.
268
269 Outputs:
270 R4 contains the address of the buffer's CCB
271 C-Bit is CLEAR if the CCB/SDB was successfully allocated
272 C-Bit is SET if the allocation failed
273
274 Note:
275 APR6 is mapped to the allocated buffer on exit.
276
277 .ENABL LSB
278
279 000302 $CSBGT::SAVRG <R2,R3,R4> ; Save some registers
280 000310 000241 CLC ; Assume success
281 000312 INHIB$;;; Inhibit Interrupts (and save C-Bit)
282
283 .IF DF M$$PRO
284 CALL $MPLCK ;;; Lock access to CommExec resources
285 .ENDC
286
287 000324 CALL CCBGT ;;; Allocate a CCB
288 000330 103444 BCS 20$;;; If CS, CCB allocation failed
289
290 000332 012764 100014 000022 MOV #CF.LB!CF.SOM!CF.EOM,C.FLG(R4)
291 ;;; Set last buffer and single message in
292 000340 112764 000010 000003 MOVB #CB.SDB,C.BID(R4) ;;; in flags and load ID of CCB/SDB
293 000346 016764 000000G 000020 MOV $SDBSZ,C.CNT(R4) ;;; Load buffer size
294 000354 016764 000000G 000030 MOV $SDBSZ,C.CNT2(R4)
295 000362 012705 000000G MOV #SDBLH,R5 ;;; Point to Listhead in allocation ctrl block
296 000366 CALL BUFGT ;;; Allocate the buffer
297 000372 103421 BCS 10$;;; If CS, SDB allocat on failed, return CCB
298 000374 00042C BR 20$;;; Else, enter common code

```

CEBUF      CREATED BY    MACRO    ON 28-JUN-85 AT 18:17      PAGE 1      L 2

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE      | REFERENCES                                                                                       |
|--------|------------|--------------------------------------------------------------------------------------------------|
| BUFGT  | 000610 R   | 8-173      11-296      #18-578                                                                   |
| BUFRT  | 000670 R   | 12-328      #19-617                                                                              |
| CB.CCB | = 000002   | 15-473                                                                                           |
| CB.RDB | = 000004   | 8-169                                                                                            |
| CB.SDB | = 000010   | 11-292                                                                                           |
| CCBGT  | 000534 R   | 6-100      8-165      11-287      #15-436                                                        |
| CCBRT  | 000572 R   | 8-176      10-249      12-330      #16-501                                                       |
| CF.EOM | = 000004   | 11-290                                                                                           |
| CF.LB  | = 100000   | 8-168      11-290                                                                                |
| CF.SOM | = 000010   | 11-290                                                                                           |
| CMNRTN | 000442 R   | 9-228      #12-336                                                                               |
| C.BID  | 000003     | *8-169      *11-292      *15-473                                                                 |
| C.BUF  | 000014     | *9-213      12-326      12-327      *18-586      *18-588                                         |
| C.BUF2 | 000024     | *9-213      *18-587      *18-589                                                                 |
| C.CNT  | 000020     | *8-170      *11-293                                                                              |
| C.CNT2 | 000030     | *8-171      *11-294                                                                              |
| C.FLG  | 000022     | *8-168      *11-290                                                                              |
| C.FNC  | 000010     | *9-220                                                                                           |
| FAIL   | 000660 R   | 15-483      #18-596                                                                              |
| FC.RCE | = 000002   | 9-220                                                                                            |
| FS.RTN | = 001000   | 9-220                                                                                            |
| KISAR6 | = ***** GX | 8-156      *8-185      *18-582      19-618      *19-619      *19-622                             |
| LF.ACT | = 100000   | #5-63                                                                                            |
| LF.BRO | = 000400   | #5-63                                                                                            |
| LF.BWT | = 000007   | #5-63      14-399                                                                                |
| LF.ENA | = 002000   | #5-63                                                                                            |
| LF.LPB | = 001000   | #5-63                                                                                            |
| LF.MDC | = 000100   | #5-63                                                                                            |
| LF.MFL | = 004000   | #5-63                                                                                            |
| LF.MTP | = 000020   | #5-63                                                                                            |
| LF.PAC | = 000200   | #5-63                                                                                            |
| LF.RDY | = 040000   | #5-63                                                                                            |
| LF.REA | = 010000   | #5-63                                                                                            |
| LF.SER | = 000040   | #5-63                                                                                            |
| LF.TIM | = 000010   | #5-63                                                                                            |
| LF.UNL | = 020000   | #5-63                                                                                            |
| LF.X2P | = 000000   | #5-63                                                                                            |
| LN.CLO | = 000000   | #5-63                                                                                            |
| LN.DUM | = 000005   | #5-63                                                                                            |
| LN.LOA | = 000004   | #5-63                                                                                            |
| LN.LOD | = 000003   | #5-63                                                                                            |
| LN.OAU | = 000003   | #5-63                                                                                            |
| LN.OFF | = 000001   | #5-63                                                                                            |
| LN.ON  | = 000000   | #5-63                                                                                            |
| LN.OOP | = 000004   | #5-63                                                                                            |
| LN.OPE | = 000001   | #5-63                                                                                            |
| LN.REF | = 000002   | #5-63                                                                                            |
| LN.SER | = 000002   | #5-63                                                                                            |
| LN.STA | = 000017   | #5-63                                                                                            |
| LN.SUB | = 000360   | #5-63                                                                                            |
| LN.TRI | = 000006   | #5-63                                                                                            |
| LEESIT | = *****    | 6-94      6-106      8-159      8-183      9-211      10-243      10-255      11-281      12-318 |

CEBUF      CREATED BY    MACRO    ON 28-JUN-85 AT 18:17      PAGE 2      M 2

SYMBOL CROSS REFERENCE      CREF    04.00

```

258 .SBTTL $CSBGT - Get a CCB and a Small Data Buffer
259
260 *--$CSBGT- Get a CCB and a Small Data Buffer
261
262 This routine is called to allocate a CCB and a Small Data Buffer.
263 The routine also sets up the buffer descriptor in the CCB to point
264 to the buffer as well as the proper byte count.
265
266 Inputs:
267 None.
268
269 Outputs:
270 R4 contains the address of the buffer's CCB
271 C-Bit is CLEAR if the CCB/SDB was successfully allocated
272 C-Bit is SET if the allocation failed
273
274 Note:
275 APR6 is mapped to the allocated buffer on exit.
276
277 .ENABL LSB
278
279 000302 $CSBGT::SAVRG <R2,R3,R5> ; Save some registers
280 000310 000241 CLC ; Assume success
281 000312 INHIB$;;; Inhibit Interrupts (and save C-Bit)
282
283 .IF DF M$$PRO
284 CALL $MPLCK ;;; Lock access to CommExec resources
285 .ENDC
286
287 000324 CALL CCBGT ;;; Allocate a CCB
288 000330 103444 BCS 20$;;; If CS, CCB allocation failed
289
290 000332 012764 100014 000022 MOV #CF.LB!CF.SOM!CF.EOM,C.FLG(R4)
291
292 000340 112764 000010 000003 MOVB #CB.SDB,C.BID(R4) ;;; Set last buffer and single message in
293 000346 016764 000000G 000020 MOV $SDBS2,C.CNT(R4) ;;; in flags and load ID of CCB/SDB
294 000354 016764 000000G 000030 MOV $SDBS7,C.CNT2(R4) ;;; Load buffer size
295 000362 012705 000000G MOV #SDBLH,R5 ;;; Point to listhead in allocation ctrl block
296 000366 CALL BUFGT ;;; Allocate the buffer
297 000372 103421 BCS 10$;;; If CS, SDB allocation failed, return CCB
298 000374 000422 BR 20$;;; Else, enter common code

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL | VALUE      | REFERENCES                                                |
|--------|------------|-----------------------------------------------------------|
| BUFGT  | 000766 R   | 9-173 12-296 #19-578                                      |
| BUFRT  | 001046 R   | 13-328 #20-617                                            |
| CB.CCB | = 000002   | 16-473                                                    |
| CB.RDB | = 000004   | 9-169                                                     |
| CB.SDB | = 000010   | 12-292                                                    |
| CCBGT  | 000534 R   | 7-100 9-165 12-287 #16-436                                |
| CCBRT  | 000656 R   | 9-176 11-249 13-330 #17-501                               |
| CF.EOM | = 000004   | 12-290                                                    |
| CF.LB  | = 100000   | 9-168 12-290                                              |
| CF.SOM | = 000010   | 12-290                                                    |
| CMNRTN | 000442 R   | 10-228 #13-336                                            |
| C.BID  | 000003     | *9-169 *12-292 *16-473                                    |
| C.BUF  | 000014     | *10-213 *13-326 *19-586 *19-588                           |
| C.BUF2 | 000024     | 10-213 *19-587 *19-589                                    |
| C.CNT  | 000020     | *9-170 *12-293                                            |
| C.CNT2 | 000030     | *9-171 *12-294                                            |
| C.FLG  | 000022     | *9-168 *12-290                                            |
| C.FNC  | 000010     | *10-220                                                   |
| FAIL   | 001036 R   | 16-483 #19-596                                            |
| FC.RCE | = 000002   | 10-220                                                    |
| FS.RTN | = 001000   | 10-220                                                    |
| KISAR6 | = ***** GX | 9-156 *9-185 *19-582 20-618 *20-619 *20-622               |
| LF.ACT | = 100000   | #6-63                                                     |
| LF.BRO | = 000400   | #6-63                                                     |
| LF.BWT | = 000007   | #6-63 15-399                                              |
| LF.ENA | = 002000   | #6-63                                                     |
| LF.LPB | = 001000   | #6-63                                                     |
| LF.MDC | = 000100   | #6-63                                                     |
| LF.MFL | = 004000   | #6-63                                                     |
| LF.MTP | = 000020   | #6-63                                                     |
| LF.PAC | = 000200   | #6-63                                                     |
| LF.RDY | = 040000   | #6-63                                                     |
| LF.REA | = 010000   | #6-63                                                     |
| LF.SER | = 000040   | #6-63                                                     |
| LF.TIM | = 000010   | #6-63                                                     |
| LF.UNL | = 020000   | #6-63                                                     |
| LF.X2P | = 000000   | #6-63                                                     |
| LN.CLO | = 000000   | #6-63                                                     |
| LN.DUM | = 000005   | #6-63                                                     |
| LN.LOA | = 000004   | #6-63                                                     |
| LN.LOO | = 000003   | #6-63                                                     |
| LN.OAU | = 000003   | #6-63                                                     |
| LN.OFF | = 000001   | #6-63                                                     |
| LN.ON  | = 000000   | #6-63                                                     |
| LN.OOP | = 000004   | #6-63                                                     |
| LN.OPE | = 000001   | #6-63                                                     |
| LN.REF | = 000002   | #6-63                                                     |
| LN.SER | = 000002   | #6-63                                                     |
| LN.STA | = 000017   | #6-63                                                     |
| LN.SUB | = 000360   | #6-63                                                     |
| LN.TRI | = 000006   | #6-63                                                     |
| L\$SI1 | = *****    | 7-94 7-106 9-159 9-183 10-211 11-243 11-255 12-281 13-318 |



```

293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315

.ENDC

BIT #LF.MDC,@(R2)+ ; DOES LINE NEED MODEM CONTROL?
BEQ 20$; NO - JOIN COMMON CODE TO DISPATCH TO DDM
MOV R5,-(SP) ; SAVE REGISTER
MOV R3,-(SP) ; SAVE ADDRESS OF FUNCTION CODE
CLR R2 ; GET AUXILIARY PROCESS PDX INDEX (ALWAYS 0)
MOV @SPDVTA,R5 ; GET ADDRESS OF AUXILIARY PROCESS' PDV
; THAT WORKS SINCE IT'S PDV INDEX = 0
MOV Z,DAT(R5),R5 ; GET ADDRESS OF DATA BASE DESCRIPTOR BLOCK
CALL $PDDSP ; DISPATCH TO MODEM CONTROLLER
MOV (SP)+,R3 ; RESTORE R3

MOVB C,LIN(R4),R2 ; OBTAIN LINE NUMBER (NO SIGN EXTENSION)
BIS #100000,C,LIN(R4) ; INDICATE MODEM CONTROL REQUEST
CALL DDMDSP ; DISPATCH TO THE DEVICE DRIVER (DDM)
BCS 25$; REQUEST WILL COMPLETE ASYNCHRONOUSLY
CALL $CCBRT ; OTHERWISE RETURN CCB TO POOL
CLR R4 ; MODEM CONTROLLER WILL RETURN ASYNCHRONOUS COMPLETION
SEC ; SET C-BIT, LINE IS MODEM CONTROLLED
BR 25$; JOIN COMMON CODE

.ENDC

```

CEDDM      CREATED BY MACRO ON 28-JUN-85 AT 18:18      PAGE 4      L 6  
SYMBOL CROSS REFERENCE      CREF 04.00

| SYMBOL  | VALUE      | REFERENCES          |
|---------|------------|---------------------|
| \$DDXON | 000116 RG  | #7-191              |
| \$LLCSP | 000374 RG  | #13-512             |
| \$PDDSP | = ***** GX | 11-419              |
| \$PDQU1 | = ***** GX | 13-527              |
| \$PDVTA | = ***** GX | 11-399              |
| \$SLTMA | = ***** GX | 14-571              |
| \$STDDM | 000404 RG  | #14-558             |
| \$STDD1 | 000410 RG  | 11-394      #14-564 |

374

.DSABL LSB

\*\*FILE\*\*ID\*\*CEDDMN

L 8

```
CCCCCCCC EEEEEEEEE DDDDDDDD DDDDDDDD MM MM NN NN
CCCCCCCC EEEEEEEEE DDDDDDDD DDDDDDDD MM MM NN NN
CC EE DD DD DD DD MMMM MMMM NN NN
CC EE DD DD DD DD MMMM MMMM NN NN
CC EE DD DD DD DD MM MM MM NNNN NN
CC EEEEEEE DD DD DD DD MM MM NN NN NN
CC EEEEEEE DD DD DD DD MM MM NN NN NN
CC EE DD DD DD DD MM MM NN NNNN
CC EE DD DD DD DD MM MM NN NNNN
CC EE DD DD DD DD MM MM NN NN
CC EE DD DD DD DD MM MM NN NN
CCCCCCCC EEEEEEEEE DDDDDDDD DDDDDDDD MM MM NN NN
CCCCCCCC EEEEEEEEE DDDDDDDD DDDDDDDD MM MM NN NN
.....
.....
.....
.....
```

```
LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL S:SSS TT
LL S:SSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
```

```

424 .SBTTL $DDAST - ASYNCHRONOUS COMPLETION TO DLC LEVEL
425
426 ;+
427 **-$DDAST-ASYNCHRONOUS COMPLETION TO DATA LINK CONTROL
428
429 THIS SUBROUTINE IS CALLED BY DEVICE DRIVERS WHEN AN ASYNCHRONOUS
430 CONDITION MUST BE RETURNED TO A DATA LINK CONTROL MODULE AND NO CCB
431 IS AVAILABLE AT THE DEVICE DRIVER.
432
433 INPUTS:
434
435 R3 = ASYNCHRONOUS COMPLETION STATUS (MOVED INTO C.STS)
436 R4 = SYSTEM LINE NUMBER
437
438 OUTPUTS:
439
440 A CCB IS ALLOCATED ON BEHALF OF THE DEVICE DRIVER
441 AND THE ASYNCHRONOUS STATUS IS QUEUED TO THE
442 DATA LINK CONTROL MODULE WITH:
443 C.FNC= FC.CCP
444 C.MOD= FS.AST
445
446 REGISTERS MODIFIED:
447
448 XXX
449 ; -
450 .ENABL LSB
451
452 $DDAST: MOV R4, -(SP) ; SAVE LINE NUMBER
453 CALL $CCBGT ; ALLOCATE A CCB
454 BCS 5$; IF CS ERROR
455 MOV (SP)+, C.LIN(R4) ; SET LINE NUMBER IN CCB
456 MOV #FC.CCP+FS.AST, C.FNC(R4) ; SET ERROR COMPLETE FUNCTION CODE
457 BR 20$; FINISH IN COMMON CODE
458
459 5$: MOV (SP)+, R4 ; CLEAN THE STACK
460 RETURN ; RETURN TO CALLER
461

```

```

67 .SBTTL $STDLC - SET UP DLC PARAMETERS FOR A GIVEN SLN
68
69 +
70 **-$STDLC-COMPUTE DLC PARAMETERS FROM A SYSTEM LINE NUMBER
71 **-$STDLC1-(ALTERNATE ENTRY)
72
73 :
74 : INPUTS:
75 :
76 : R2 = SYSTEM LINE NUMBER (ALTERNATE ENTRY ONLY)
77 : R4 = ADDRESS OF A CCB WITH A VALID SLN IN C.LIN
78 : (MAIN ENTRY ONLY)
79 :
80 : OUTPUTS:
81 :
82 : R2 = PDV INDEX (WORD OFFSET)
83 : R3 = ADDRESS OF FUNCTION CODE CELL IN CCB
84 : R5 = ADDRESS OF DLC LINE TABLE
85 :
86 : REGISTERS MODIFIED:
87 :
88 : NONE
89 :
90 :
91 $STDLC::IF NDF N$$1LN
92 MOV C.LIN(R4),R2 ; GET SYSTEM LINE NUMBER
93
94 .ENDC
95
96 $STDLC1::IF DF N$$1LN
97 MOV @$$SLTMA,R2 ; GET ADDRESS OF SYSTEM LINE TABLE ENTRY
98
99 .IFF
100
101 ASL R2 ; FORM WORD INDEX
102 ADD $$SLTMA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE
103 MOV (R2),R2 ; GET ADDRESS OF SYSTEM LINE TABLE ENTRY
104
105 .ENDC
106
107 MOV L.DLS(R2),R5 ; GET LINE TABLE ADDRESS
108 MOV L.DLC(R2),R2 ; GET PDV INDEX (WORD INDEX)
109 MOV R4,R3 ; COMPUTE A POINTER TO THE FUNCTION CODE
110 ADD #C.FNC,R3 ;
111 RETURN ; RETURN TO CALLER
112
113

```

```
58
59
60 000000
61 000000
62 000000
63
 .MCALL CCBDF$,PDVDF$,SAVRG,RESRG,SLTDF$
 .MCALL CALLR ; AVOID SYSTEM DEPENDENCY
 CCBDF$; DEFINE THE CCB OFFSETS
 PDVDF$; DEFINE THE PDV OFFSETS
 SLTDF$; DEFINE THE SLT OFFSETS
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54

```
.IIF NDF N$$EVL .TITLE CELOG - COMM/EXEC EVENT LOGGING ROUTINES
.IIF DF N$$EVL .TITLE CELOG1 - COMM/EXEC EVENT LOGGING ROUTINES
.IDENT /V05.00/
```

```
: COPYRIGHT (C) 1980, 1982, 1983, 1985 BY
: DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
```

```
: THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
: SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
: INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR
: ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE
: MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH
: SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE
: TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN
: IN DEC.
```

```
: THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
: NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
: EQUIPMENT CORPORATION.
```

```
: DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
: ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
```

MODULE DESCRIPTION

CEX EVENT LOGGING ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 30-JAN-80
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/R SX V1.0

.MCALL PDVDF\$.SAVRG,RESRG

PDVDF\$ ; DEFINE PDV OFFSETS

000000



CELOG1      CREATED BY    MACRO    ON 28-JUN-85 AT 18:20

PAGE 2      L 13

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|         |       |       |
|---------|-------|-------|
| CALL    | 6-117 |       |
| PDVDF\$ | #5-52 | 5-54  |
| RESRG   | #5-52 | 6-118 |
| RETURN  | 6-125 |       |
| SAVRG   | #5-52 | 6-114 |

```

352
353
354
355
356
357
358
359
360
361 000304
362
363
364
365
366 000304 012703 000000G
367 000310 005002
368 000312
369
370
371
372
373
374 000316
375
376 000001

.SBTTL POWERFAIL RECOVERY ROUTINE
;+
;***$CEPWR-POWERFAIL RECOVERY ROUTINE
;
;This routine is dispatched from the executive powerfail recovery
;code. It will dispatch to the AUX process with a powerfail
;recovery function code.
;-
$CEPWR::IF DF R$$MPL
MOV $DDFNC,-(SP) ; Save current function code
MOV #FC.PWR,$DDFNC ; Set up powerfail function code
.ENDC
MOV #DDFNC,R3 ; Point to powerfail recovery function code
CLR R2 ; Set up PDV index (AUX is always zero)
CALL $PD SPL ; and dispatch to the process
IF DF R$$MPL
MOV (SP)+,$DDFNC ; Restore current function code
.ENDC
RETURN
.ENDC

```

CESUB MACRO V05.03b Friday 28-Jun-85 18:20 Page 7-1  
\$CNV18 - CONVERT TO 18-BIT UNIBUS ADDRESS

L 15

199 000166 005502  
200 000170 010203  
201 000172 012602  
202 000174  
203

10\$:

ADC R2  
MOV R2,R3  
MOV (SP)+,R2  
RETURN  
.DSABL LSB

; ADD OVERFLOW TO BITS 16 & 17  
; GET HIGH ORDER BITS INTO R3  
; AND LOW ORDER BITS INTO R2  
; RETURN

CESUB      CREATED BY MACRO ON 28-JUN-85 AT 18:20      PAGE 2      L 16  
SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL  | VALUE                | REFERENCES                                 |
|---------|----------------------|--------------------------------------------|
| \$DIV   | =        *****    GX | 14-511                                     |
| \$MUL   | =        *****    GX | 13-477                                     |
| \$MVFBF | 000242    RG         | #9-293                                     |
| \$MVTBF | 000176    RG         | #8-244                                     |
| \$PDVID | 000514    RG         | #12-430                                    |
| \$PDVNM | =        *****    GX | 12-432                                     |
| \$PDVTA | =        *****    GX | 10-335      10-348      12-431      12-443 |
| \$PUMR  | =        *****    GX | 7-181                                      |
| \$XBIAS | =        *****    GX | 11-390                                     |

\$CSBRT - Return a CCB and a Small Data Buffer

```

300 .SBTTL $CSBRT - Return a CCB and a Small Data Buffer
301 *
302 **-$CSBRT- Return a CCB and a Small Data Buffer
303 :
304 : This routine is called to deallocate a CCB and a Small Data Buffer
305 : which is pointed to by the first buffer descriptor in the CCB.
306 :
307 : Inputs:
308 : R4 contains the address of the CCB
309 : Note that C.BUF+2 must contain the original Virtual
310 : Address of the buffer, and C.BUF must contain the original
311 : Bias of the buffer.
312 :
313 : Outputs:
314 : None.
315 :
316 :-
317 000376 $CSBRT::SAVRG <R2,R3,R5> ; Save some registers
318 000404 INHIB$;;; Inhibit Interrupts
319
320 000416 012705 000000G MOV #SDBCT,R5 ;;; Point to count in allocation control block
321
322 000422 RELBUF: .IF DF M$$PRO
323 : CALL $MPLCK ;;; Lock access to CommExec resources
324 : .ENDC
325
326 000422 016403 000014 MOV C.BUF(R4),R3 ;;; Set up address of the buffer
327 000426 016402 000016 MOV C.BUF+2(R4),R2 ;;; as an address double word
328 000432 CALL BUFRT ;;; Release the buffer
329
330 000436 10$: CALL CCBRT ;;; Release the CCB
331
332 000442 20$: .IF DF M$$PRO
333 : CALL @($P)+ ;;; Co-Routine return to unlock resources
334 : .ENDC
335
336 000442 CMNRTN: ENABL$; Enable interrupts (and load final C-Bit)
337 000446 RESRG <R5,R3,R2> ; Restore registers
338 000454 RETURN
339
340 .DSABL LSR

```

\$RDBW\* - Queue a request for a Receive Data Buffer

| SYMBOL  | VALUE       | REFERENCES                                             |
|---------|-------------|--------------------------------------------------------|
| L.COST  | 000015      | 12-336                                                 |
| L.CTL   | 000012      | #5-63                                                  |
| L.CVA   | 177776      | #5-63                                                  |
| L.DDM   | 000002      | #5-63                                                  |
| L.DDS   | 000004      | #5-63                                                  |
| L.DLC   | 000003      | 45-63                                                  |
| L.DLM   | 000006      | #5-63                                                  |
| L.DLS   | 000010      | #5-63                                                  |
| L.FLG   | 000000      | #5-63                                                  |
| L.KRBA  | 000016      | #5-63                                                  |
| L.LEN   | = 000022    | #5-63                                                  |
| L.MPF   | 000022      | #5-63                                                  |
| L.NMST  | 000020      | #5-63                                                  |
| L.NSTA  | 000014      | #5-63                                                  |
| L.OWNR  | 000021      | #5-63                                                  |
| L.UNT   | 000013      | #5-63                                                  |
| M\$SPRO | = *****     | 5-71 6-96 6-102 8-161 8-179 9-222 10-245 10-251 11-283 |
|         |             | 12-322 12-332 13-360 13-369 14-392 14-403 15-475       |
| N\$SOPT | = *****     | 4-1 4-2 15-436 16-501                                  |
| N\$SVCT | = *****     | 8-156 8-185 18-582 19-618 19-619 19-622                |
| PRIOFF  | = 000000    | #5-69 *18-596                                          |
| PR7     | = *****     | 6-94 8-159 9-211 10-243 11-281 12-318                  |
| PS      | = *****     | 6-94 *6-106 8-159 *8-159 *8-183 9-211 *9-211 10-243    |
|         |             | *10-243 *10-255 11-281 *11-281 12-318 *12-318 *12-336  |
| RELBUF  | 000422 R    | 9-217 #12-322                                          |
| SF.ACT  | = 000200    | #5-63                                                  |
| SF.ENA  | = 000100    | #5-63                                                  |
| SF.LPB  | = 000004    | #5-63                                                  |
| SF.MFL  | = 000040    | #5-63                                                  |
| SF.PAC  | = 000020    | #5-63                                                  |
| SF.REA  | = 000010    | #5-63                                                  |
| SF.SER  | = 000001    | #5-63                                                  |
| SF.SVC  | = 000002    | #5-63                                                  |
| SF.UNL  | = 000040    | #5-63                                                  |
| S.COST  | 000001      | #5-63                                                  |
| S.FLG   | 000000      | #5-63                                                  |
| S.LEN   | 000004      | #5-63                                                  |
| S.NMST  | 000002      | #5-63                                                  |
| S.OWNR  | 000003      | #5-63                                                  |
| \$CCBAF | = ***** GX  | *15-482                                                |
| \$CCBCT | = ***** GX  | *15-471                                                |
| \$CCBGT | = 000000 RG | *16-534 #6-93                                          |
| \$CCBLH | = ***** GX  | 15-468 *15-470 16-532 *16-533                          |
| \$CLBRT | 000256 RG   | #10-243                                                |
| \$CSBGT | 000302 RG   | #11-279                                                |
| \$CSBRT | 000375 RG   | #12-317                                                |
| \$LDBAF | = ***** GX  | *7-134                                                 |
| \$LDBGT | 000026 RG   | *7-132                                                 |
| \$LDBRT | 000170 RG   | *9-209                                                 |
| \$PDGUE | = ***** GX  | 9-227                                                  |
| \$RDBCT | = ***** GX  | 7-132 9-215                                            |

\$CSBRT - Return a CCB and a Small Data Buffer

```

300 .SBTTL $CSBRT - Return a CCB and a Small Data Buffer
301 ;+
302 **-$CSBRT- Return a CCB and a Small Data Buffer
303
304 This routine is called to deallocate a CCB and a Small Data Buffer
305 which is pointed to by the first buffer descriptor in the CCB.
306
307 Inputs:
308 R4 contains the address of the CCB
309 Note that C.BUF+2 must contain the original Virtual
310 Address of the buffer, and C.BUF must contain the original
311 Bias of the buffer.
312
313 Outputs:
314 None.
315 :-
316
317 000376 $CSBRT::SAVRG <R2,R3,R5> ; Save some registers
318 000404 INHIB$;;; Inhibit Interrupts
319
320 000416 012705 000000G MOV #$$DBCT,R5 ;;; Point to count in allocation control block
321
322 000422 RELBUF: .IF DF M$$PRO
323 CALL $MPLCK ;;; Lock access to CommExec resources
324 .ENDC
325
326 000422 016403 000014 MOV C.BUF(R4),R3 ;;; Set up address of the buffer
327 000426 016402 000016 MOV C.BUF+2(R4),R2 ;;; as an address double word
328 000432 CALL BUFRT ;;; Release the buffer
329
330 000436 10$: CALL CCBRT ;;; Release the CCB
331
332 000442 20$: .IF DF M$$PRO
333 CALL @($P)+ ;;; Co-Routine return to unlock resources
334 .ENDC
335
336 000442 CMNRTN: ENAB$; Enable interrupts (and load final C-Bit)
337 000446 RESRG ; Restore registers
338 000454 RETURN
339
340 .DSABL LSB

```

\$RDBWT - Queue a request for a Receive Data Buffer

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE    | REFERENCES                                                       |
|---------|----------|------------------------------------------------------------------|
| L.COST  | 000015   | 13-336                                                           |
| L.CTL   | 000012   | #6-63                                                            |
| L.CVA   | 177776   | #6-63                                                            |
| L.DDM   | 000002   | #6-63                                                            |
| L.DDS   | 000004   | #6-63                                                            |
| L.DLC   | 000003   | #6-63                                                            |
| L.DLM   | 000006   | #6-63                                                            |
| L.DLS   | 000010   | #6-63                                                            |
| L.FLG   | 000000   | #6-63                                                            |
| L.KRBA  | 000016   | #6-63                                                            |
| L.LEN   | = 000022 | #6-63                                                            |
| L.MPF   | 000022   | #6-63                                                            |
| L.NMCT  | 000020   | #6-63                                                            |
| L.NSTA  | 000014   | #6-63                                                            |
| L.OWNR  | 000021   | #6-63                                                            |
| L.UNT   | 000013   | #6-63                                                            |
| M\$PRO  | = *****  | 6-71 7-96 7-102 9-161 9-179 10-222 11-245 11-251 12-283          |
|         |          | 13-322 13-332 14-360 14-369 15-392 15-403 16-450 16-459 16-475   |
|         |          | 17-520                                                           |
| N\$SOPT | = 000001 | #4-2 5-1 5-2 16-436 17-501                                       |
| N\$SVCT | = *****  | 9-156 9-185 19-582 20-618 20-622                                 |
| PRIOFF  | = 000000 | #6-69 16-438 17-508 *19-596                                      |
| PR7     | = *****  | 9-159 10-211 11-243 12-281 13-318                                |
| PS      | = *****  | 7-94 7-94 *7-94 *7-106 9-159 *9-159 *9-183 10-211 *10-211 11-243 |
|         |          | *11-243 *11-255 12-281 *12-281 13-318 *13-318 *13-336            |
| RELBUF  | 000422 R | 10-217 #13-322                                                   |
| R\$MPL  | = *****  | 17-503                                                           |
| SF.ACT  | = 000200 | #6-63                                                            |
| SF.ENA  | = 000100 | #6-63                                                            |
| SF.LPB  | = 000004 | #6-63                                                            |
| SF.MFL  | = 000040 | #6-63                                                            |
| SF.PAC  | = 000020 | #6-63                                                            |
| SF.REA  | = 000010 | #6-63                                                            |
| SF.SER  | = 000001 | #6-63                                                            |
| SF.SVC  | = 000002 | #6-63                                                            |
| SF.UNL  | = 000040 | #6-63                                                            |
| S.COST  | 000001   | #6-63                                                            |
| S.FLG   | 000000   | #6-63                                                            |
| S.LEN   | 000004   | #6-63                                                            |
| S.NMST  | 000002   | #6-63                                                            |
| S.OWNR  | 000003   | #6-63                                                            |
| \$ALOCB | = *****  | GX 16-454                                                        |
| \$CCBAF | = *****  | GX *16-482                                                       |
| \$CCBAL | = *****  | GX 16-444 *16-463 17-513 *17-527                                 |
| \$CCBCT | = *****  | GX 16-441 *16-471 17-511 *17-534                                 |
| \$CCBGT | 000000   | RG #7-93                                                         |
| \$CCBLH | = *****  | GX 16-468 *16-470 17-532 *17-533                                 |
| \$CCBRT | 000256   | RG #11-243                                                       |
| \$CCBSZ | = *****  | GX 16-448 17-518                                                 |
| \$CEAVL | = *****  | GX 17-504                                                        |
| \$CSBGT | 000302   | RG #12-279                                                       |



```

317 .SBTTL DDCM3 - COMMON PROCESS FOR FUNCTIONS WITH CCB
318 .SBTTL DDCM4 - COMMON PROCESS FOR FUNCTIONS WITH CCB
319
320 *
321 **DDCM3 - COMMON PROCESS FOR FUNCTIONS WITH CCB
322 **DDCM4 - COMMON PROCESS FOR FUNCTIONS WITH CCB
323
324 INPUTS:
325 R3 = ADDRESS OF FUNCTION AND SUBFUNCTION CODE (DDCM3)
326 R3 = ADDRESS OF FUNCTION CODE ONLY (DDCM4)
327 R4 = ADDRESS OF CCB
328 STACK CONTAINS:
329 00(SP) = CALLERS R3
330 02(SP) = RETURN ADDRESS TO CALLING DLC
331
332 OUTPUTS TO DDM:
333 R3 = SUBFUNCTION CODE (WORD INDEX)
334 R4 = ADDRESS OF CCB
335 R5 = ADDRESS OF LINE TABLE
336
337 REGISTERS ACROSS CALL:
338 R2,R3,R5 - PRESERVED
339 R4 - MAY BE MODIFIED
340
341
342 000150 DDCM3: .IF DF K$$DAS
343
344 MOV R3,$DDFNC ; STORE FUNCTION CODE IN DATA AREA
345 MOV #DDFNC,R3 ; POINT R3 AT FUNCTION CODE
346
347 .IFTF ; DF K$$DAS
348
349 MOV (R3),C.FNC(R4) ; PUT FUNCTION AND SUBFUNCTION IN CCB
350 000150 011364 000010 BR 10$; JOIN COMMON CODE
351 000154 000402
352
353 000156 DDCM4: .IFT ; DF K$$DAS
354
355 MOV R3,$DDFNC ; STORE FUNCTION CODE IN DATA AREA
356 MOV #DDFNC,R3 ; POINT R3 AT FUNCTION CODE
357
358 .ENDC ; DF K$$DAS
359
360 000156 111364 000010 MOV (R3),C.FNC(R4) ; PUT FUNCTION CODE IN CCB
361
362 000162 010246 10$: MOV R2,-(SP) ; SAVE R2
363
364 000164 116402 000006 20$: MOV C.LIN(R4),R2 ; GET SLN
365
366 000170 010546 22$: MOV R5,-(SP) ; SAVE R5
367 CALL DDMDSP ; DISPATCH TO THE DEVICE DRIVER (DDM)
368 000172 012605 25$: MOV (SP)+,R5 ; RESTORE R5
369 000176 012605 MOV (SP)+,R2 ; RESTORE R2
370 000200 012602 MOV (SP)+,R3 ; RESTORE R3
371 000202 012603 RETURN ; RETURN TO CALLING DLC
372 000204
373

```

CEDDM      CREATED BY MACRO ON 28-JUN-85 AT 18:18      PAGE 5      M 6  
 MACRO CROSS REFERENCE      CREF      04.00

| MACRO NAME | REFERENCES |        |        |        |       |       |       |       |       |       |
|------------|------------|--------|--------|--------|-------|-------|-------|-------|-------|-------|
| CALL       | 10-368     | 11-394 | 11-419 | 12-453 |       |       |       |       |       |       |
| CALLR      | #4-65      | 13-508 | 13-527 |        |       |       |       |       |       |       |
| CCBDF\$    | #4-64      | 4-66   |        |        |       |       |       |       |       |       |
| DDFDF      | #5-97      | 7-174  | 7-175  | 7-177  | 7-179 | 7-180 | 7-181 | 7-182 | 7-183 | 7-184 |
|            | 7-185      | 7-187  | 7-188  | 7-189  | 7-191 | 7-192 | 7-193 |       |       |       |
| ENABL\$    | #4-63      |        |        |        |       |       |       |       |       |       |
| INHIB\$    | #4-63      |        |        |        |       |       |       |       |       |       |
| MFPS       | 11-396     |        |        |        |       |       |       |       |       |       |
| MTPS       | 11-404     | 11-421 |        |        |       |       |       |       |       |       |
| PCVDF\$    | #4-64      | 4-67   |        |        |       |       |       |       |       |       |
| RESRG      | #4-63      | 11-418 |        |        |       |       |       |       |       |       |
| RETURN     | 10-372     | 11-422 | 12-460 | 14-578 |       |       |       |       |       |       |
| SAVRG      | #4-63      | 11-397 |        |        |       |       |       |       |       |       |
| SLTDF\$    | #4-64      | 4-68   |        |        |       |       |       |       |       |       |

```

376 .SBTTL DDMDSP - DISPATCH TO DDM LEVEL
377
378 ;+
379 **--DDMDSP-DISPATCH TO DDM LEVEL
380 ;
381 THIS SUBROUTINE IS CALLED TO DISPATCH DOWN TO THE DDM LEVEL. PROCESS
382 DISPATCH WILL OCCUR AT THE CURRENT PRIORITY OR AT THE PROCESS PRIORITY
383 DEPENDENT ON A PDV FLAG.
384 ;
385 INPUTS:
386 ;
387 R2 = SYSTEM LINE NUMBER
388 ;
389 REGISTERS MODIFIED:
390 ;
391 R5
392 ;
393 -
394 DDMDSP: CALL $STD1 ; SET UP DDM PDV INDEX & LINE TABLE ADDRESS
395 CLC ; SAVE CURRENT PRIORITY
396 MFPS -(SP) ; WITH C-BIT CLEAR
397 SAVRG <R0> ; GET A FREE REGISTER
398 MOV R2,R0 ; COPY DDM PDV INDEX
399 ADD $PDVTA,R0 ; POINT INTO PDV INDEX TABLE
400 MOV (R0),R0 ; GET POINTER TO PDV
401 BIT #ZF.DVP,Z.FLG(R0)
402 BEQ 10$; IF EQ, RUN DDM AT CURRENT PRIORITY
403
404 MTPS #PR7 ; RAISE PROCESSOR PRIORITY
405
406 .IF DF L$$S11
407
408 MFPS -(SP) ;;; GET CURRENT PRIORITY
409 BICB Z.SCH(R0),(SP) ;;; SET UP PROCESS PRIORITY
410 MTPS (SP)+ ;;; SET UP PROCESSOR PRIORITY
411
412 .IFF
413
414 BICB Z.SCH(R0),PS ;;; DROP TO PROCESS PRIORITY
415
416 .ENDC
417
418 10$: RESRG <R0> ; RESTORE REGISTER
419 CALL $PDDSP ; DISPATCH TO THE DDM
420 ADC (SP) ; UPDATE C-BIT IN SAVED PS
421 MTPS (SP)+ ; RESTORE PROCESSOR PRIORITY
422 RETURN

```

CEDDMN MACRO V05.03b Friday 28-Jun-85 18:18  
Table of contents

15- 533 \$STDDM - SET DDM PDV INDEX AND LINE TABLE ADDRESS

```

463 .SBTTL $DDXMP - TRANSMIT COMPLETE TO DLC LEVEL
464 .SBTTL $DDRCR - RECEIVE COMPLETE TO DLC LEVEL
465 .SBTTL $DDCCP - CONTROL COMPLETE TO DLC LEVEL
466 .SBTTL $DDKCP - KILL COMPLETE TO DLC LEVEL
467
468 *--$DDXMP-TRANSMIT COMPLETE TO DATA LINK CONTROL
469 *--$DDRCR-RECEIVE COMPLETE
470 *--$DDCCP-CONTROL COMPLETE
471 *--$DDKCP-KILL COMPLETE
472 *--$LLCSP-SPECIAL ENTRY POINT FOR X25
473
474 THIS SUBROUTINE IS CALLED BY DEVICE DRIVERS TO QUEUE
475 COMPLETION NOTIFICATIONS TO DATA LINK CONTROL MODULES.
476
477 THE $LLCSP ENTRY POINT IS PROVIDED FOR USE BY THE X25 DATA LINK MAPPING.
478
479 INPUTS:
480
481 R3 = OPERATION COMPLETION STATUS
482 R4 = ADDRESS OF CCB TO QUEUE
483 THE CCB CONTAINS A VALID LINE NUMBER
484
485 OUTPUTS:
486
487 THE CCB IS QUEUED TO A DLC LIST BASED
488 ON THE SYSTEM LINE NUMBER PARAMETER IN THE CCB WITH A
489 TRANSMIT OR RECEIVE COMPLETE FUNCTION CODE.
490
491 REGISTERS MODIFIED:
492
493 XXX
494
495 -
496
497 $DDXMP::MOV B #FC.XCP,C.FNC(R4) ; SET TRANSMIT COMPLETE FUNCTION CODE
498 BR 20$; JOIN COMMON CODE
499
500 $DDRCR::MOV B #FC.RCP,C.FNC(R4) ; SET RECEIVE COMPLETE FUNCTION CODE
501 BR 20$; JOIN COMMON CODE
502
503 $DDKCP::MOV B #FC.KCP,C.FNC(R4) ; SET KILL COMPLETE FUNCTION CODE
504 BR 20$; JOIN COMMON CODE
505
506 $DDCCP::TST C,LIN(R4) ; SPECIAL MODEM CONTROL COMPLETION?
507 BPL 10$; NO
508 CALLR $CCBRT ; YES - RETURN CCB TO POOL
509 ; MODEM CONTROLLER WILL POST COMPLETION
510 10$: MOV B #FC.CCP,C.FNC(R4) ; SET CONTROL COMPLETE FUNCTION CODE
511
512 $LLCSP::
513 $DDSPC::
514 20$: MOV R3,C.STS(R4) ; SET COMPLETION STATUS
515
516 .IF DF M$$PRO
517
518 MOV B C,LIN(R4),R3 ; GET SYSTEM LINE #
519 ASL R3 ; FORM WORD OFFSET

```

\$ASCMP - ASYNCHRONOUS COMPLETION TO LLC LEVEL

```

115 .SBTTL $ASCMP - ASYNCHRONOUS COMPLETION TO LLC LEVEL
116
117
118 :+
119 :*** $ASCMP -ASYNCHRONOUS COMPLETION TO LOGICAL LINK CONTROL MODULES
120 :THIS ROUTINE IS CALLED BY DATA LINK CONTROL PROCESSES TO PASS
121 :ASYNCHRONOUS STATUS/ERROR INFORMATION TO HIGHER LEVEL PROCESSES.
122
123 :INPUTS:
124
125 :R2 = STATION NUMBER AND SYSTEM LINE NUMBER
126 :R3 = ASYNCHRONOUS STATUS/ERROR
127
128 :OUTPUTS:
129
130 :A CCB IS ALLOCATED ON BEHALF OF THE CALLER AND BASED ON THE
131 :SYSTEM LINE NUMBER. IS QUEUED TO THE APPROPRIATE LOGICAL
132 :LINK CONTROL PROCESS.
133
134 :REGISTERS MODIFIED:
135
136 :R3 AND R4
137
138 :-
139 :.ENABL LSB
140 $ASCMP::CALL $CCBGT ; ALLOCATE A CCB
141 000034 BCS 30$; IF CS ERROR
142 000040 103455 ;
143 000042 012764 000020 000010 MOV #FC.CCP+FS.AST,C.FNC(R4) ; SET ERROR COMPLETE FUNCTION CODE
144 000050 010264 000006 MOV R2,C.LIN(R4) ; SET UP STATION ADDRESS/SYSTEM LINE NUMBER
145 000054 000417 BR 10$; FINISH IN COMMON CODE

```

```

65 .SBTTL $LLCRQ - LLC TO DLC REQUEST QUEUING SUBROUTINE
66 :+
67 :*- $LLCRQ - LLC TO DLC REQUEST QUEUING SUBROUTINE
68 :
69 :THIS ROUTINE IS CALLED BY A LLC PROCESS TO QUEUE REQUESTS TO
70 :A DLC PROCESS. NOTE THAT THE CALLERS CONTEXT IS SAVED AS IF
71 :AN INTERRUPT HAD OCCURRED.
72 :
73 :INPUTS:
74 :R4= ADDRESS OF CCB (OR FIRST IN CHAIN)
75 :THE CCB MUST CONTAIN A VALID:
76 :C.FNC & C.MOD - REQUEST FUNCTION CODE
77 :C.LIN - LLC CHANNEL NUMBER (LOGICAL LINE NUMBER)
78 :
79 :OUTPUTS:
80 :THE REQUEST IS DISPATCHED TO THE PROPER DLC PROCESS
81 :
82 :REGISTERS MODIFIED:
83 :NONE
84 :-
85 000000 $LLCRQ::
86 .IF DF M$$PRO
87 :
88 SAVRG <R1,R3,R4> ; SAVE REGISTERS
89 MOV $CMPDV,R1 ; GET CURRENT LLC PDV INDEX
90 ADD $PDVTA,R1 ; POINT INTO PDV INDEX TABLE
91 MOVB C.LIN(R4),R3 ; GET LLC CHANNEL NUMBER
92 ASL R3 ; MAKE IT A WORD INDEX
93 ADD (R1),R3 ; POINT TO LLC CHANNEL MAPPING TABLE ENTRY
94 MOV Z.MAP(R3),C.LIN(R4) ; STORE SLN AND STATION ADDRESS INTO CCB
95 : NOTE - BIT15 WILL ALWAYS BE CLEAR TO
96 : FLAG DESTINATION PROCESS AS A DLC
97 :
98 MOVB C.LIN(R4),R1 ; GET SYSTEM LINE NUMBER
99 ASL R1 ; FORM WORD OFFSET
100 ADD $SLTMA,R1 ; POINT INTO SYSTEM LINE INDEX TABLE
101 MOV (R1),R1 ; GET ADDRESS OF SYSTEM LINE TABLE ENTRY
102 MOV L.KRBA(R1),R1 ; GET POINTER TO KRB
103 MOV K.URM(R1),C.URM(R4) ; SET UP UNIBUS RUN MASK
104 CALL $PDQU1 ; QUEUE REQUEST CCB AND SCHEDULE PROCESS
105 MTPS #0 ; DROP PRIORITY TO 0 AND CLEAR CONDITION CODES
106 RESRG <R4,R3,R1> ; RESTORE REGISTERS
107 RETURN
108 :
109 .IFF ; DF M$$PRO
110 :
111 SAVRG <R0,R1,R2,R3,R4,R5> ; SAVE ALL REGISTERS
112 MOV $CMPDV,R1 ; GET CURRENT LLC PDV INDEX
113 ADD $PDVTA,R1 ; POINT INTO PDV INDEX TABLE
114 MOVB C.LIN(R4),R3 ; GET LLC CHANNEL NUMBER WITHOUT SIGN EXTENSION
115 ASL R3 ; MAKE IT A WORD INDEX
116 ADD (R1),R3 ; POINT TO LLC CHANNEL MAPPING TABLE ENTRY
117 MOV Z.MAP(R3),C.LIN(R4) ; STORE SLN AND STATION ADDRESS INTO CCB
118 : NOTE - BIT15 WILL ALWAYS BE CLEAR TO
119 : FLAG DESTINATION PROCESS AS A DLC
120 CALL $STDLC ; SET UP DLC PDV INDEX AND LINE TABLE ADDRESS
121 CALL $PDSPL ; DISPATCH TO DLC

```

```

56 .SBTTL LOG NETWORK EVENT
57
58 *
59 ***$CELOG-LOG NETWORK EVENT
60
61 INPUTS:
62 R0 - EVENT CLASS AND TYPE
63 R1 - EVENT CONTROL WORD:
64 LOW BYTE:
65 BIT 0 1 => R4 IS ADDRESS OF A CCB
66 0 => R4 IS A DATA AREA POINTER
67
68 1 1 => LINE-ID IS SLN & STATION
69 0 => LINE-ID IS PDV & CHANP.L
70
71 2 1 => EVENT IS ASSOCIATED WITH A LINE
72 0 => EVENT IS NOT ASSOCIATED WITH A LINE
73
74 3 1 => EVENT IS ASSOCIATED WITH A REMOTE NODE
75 0 => EVENT IS NOT ASSOCIATED WITH A REMOTE NODE
76
77 4 1 => EVENT IS ASSOCIATE WITH A CIRCUIT
78 0 => EVENT IS NOT ASSOCIATED WITH A CIRCUIT
79
80 5 1 => EVENT IS ASSOCIATED WITH A MODULE
81 0 => EVENT IS NOT ASSOCIATED WITH A MODULE
82
83 6 1 => USE LINE-ID FROM CCB
84 0 => USE LINE-ID FROM EVENT DESCRIPTOR BLOCK
85
86 HIGH BYTE:
87 # OF BYTES OF DATA TO COPY
88
89 R3 - POINTER TO EVENT DESCRIPTOR BLOCK:
90
91 WORD 1 LINE-ID (SLN & STATION OR PDV & CHANNEL)
92 2-4 EVENT DEPENDENT DATA
93 5 NODE ADDRESS
94 6 MODULE ID (CODED VALUE)
95 7 PORT #
96 8 LOGICAL CHANNEL #
97
98 R4 - POINTER TO CCB OR DATA AREA POINTER
99
100 REGISTERS MODIFIED:
101 R0, R1, R2, R3, R5
102
103 -
104 $CELOG::IF DF N$SEVL
105
106 MOV $LGPDV,R2 ; GET PDV INDEX OF LOGGING PROCESS
107 BEQ 10$; IF EQ, NO LOGGING
108
109 MOV $CMPDV,-(SP) ; SAVE CALLING PROCESS PDV INDEX
110 MOV R3,-(SP) ; SAVE ADDRESS OF EVENT DESCRIPTOR BLOCK
111 MOV R1,-(SP) ; AND EVENT CONTROL WORD
112 MOV SP,R1 ; SET UP POINTER TO CONTROL BLOCK

```

000000



••FILE••ID••CESCH

M 13

```
CCCCCCCC EEEEEEEEE SSSSSSSS CCCCCCCC HH HH
CCCCCCCC EEEEEEEEE SSSSSSSS CCCCCCCC HH HH
CC EE SS CC HH HH
CC EE SS CC HH HH
CC EE SS CC HH HH
CC EEEEEEE SSSSSS CC HHHHHHHHHH
CC EEEEEEE SSSSSS CC HHHHHHHHHH
CC EE SS CC HH HH
CC EE SS CC HH HH
CC EE SS CC HH HH
CC EE SS CC HH HH
CCCCCCCC EEEEEEEEE SSSSSSSS CCCCCCCC HH HH
CCCCCCCC EEEEEEEEE SSSSSSSS CCCCCCCC HH HH
.....
.....
.....
.....
```

```
LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TY
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
```

N 13

|                |                 |                 |                 |                   |
|----------------|-----------------|-----------------|-----------------|-------------------|
| ASSCHK= 000000 | CP.2FR= 000030  | C.XACP 000004   | FS.SFS= 004000  | SS\$WRG= 000000   |
| ASSCPS= 000000 | CS.ABO= 000100  | C.XID 000035    | FS.SPW= 040000  | SS\$YSZ= 007600   |
| ASSPRI= 000000 | CS.BRO= 000002  | C.XLEN 000044   | FS.STM= 000000  | T\$KMG= 000000    |
| ASSTRP= 000000 | CS.BUF= 000200  | C.XPLI 000040   | FS.STP= 002000  | T\$SMIN= 000000   |
| CB.CCB= 000002 | CS.CES= 000002  | C.XPT 000034    | FS.STR= 001000  | V\$SCTR= 001000   |
| CB.DDM= 000040 | CS.CHN= 000010  | C.XSVC 000042   | FS.TRM= 003000  | X\$SDBT= 000000   |
| CB.DLC= 000020 | CS.CMP= 000200  | C.XTC 000037    | FS.WLB= 001000  | ZF.COUC= 001000   |
| CB.RDB= 000004 | CS.DCR= 000400  | C.X25 000036    | FS.XKL= 002000  | ZF.DDM= 000001    |
| CB.SDB= 000010 | CS.DEF= 000004  | D\$BUB= 177514  | FS.XOF= 010000  | ZF.DIA= 004000    |
| CB.SLI= 000100 | CS.DEV= 000002  | D\$ISK= 000000  | FS.XON= 007000  | ZF.DLC= 000002    |
| CB.XLB= 000001 | CS.DIS= 000040  | D\$SL1= 000001  | FS.ZER= 002000  | ZF.DVP= 100000    |
| CC.LLC= 000200 | CS.ENA= 000001  | D\$SYNC= 000000 | FS\$LVL= 000001 | ZF.INI= 040000    |
| CE.ABO= 100362 | CS.ENB= 000020  | D\$SYNM= 000000 | G\$STPP= 000000 | ZF.KMX= 000020    |
| CE.DAO= 100346 | CS.ERR= 100000  | E\$XPR= 000000  | G\$STSS= 000000 | ZF.LLC= 000004    |
| CE.DIS= 100366 | CS.FTL= 001000  | FC.CCP= 000020  | G\$STTK= 000000 | ZF.LMC= 000100    |
| CE.ERR= 100370 | CS.HCR= 000001  | FC.CTL= 000006  | G\$SWRD= 000000 | ZF.MAN= 020000    |
| CE.ILN= 100350 | CS.HFE= 002000  | FC.KCP= 000016  | J\$RAR= 000000  | ZF.MFL= 000010    |
| CE.LTO= 100356 | CS.LST= 040000  | FC.KIL= 000004  | J\$SRDN= 000000 | ZF.MTM= 000400    |
| CE.MOP= 100372 | CS.MTL= 004000  | FC.MAN= 000024  | KISAR5= *****   | ZF.MUX= 000040    |
| CE.NTE= 100361 | CS.RNG= 000010  | FC.MLD= 000026  | KISAR6= *****   | ZF.PSE= 002000    |
| CE.RTE= 100376 | CS.ROV= 000004  | FC.PCT= 000030  | K\$CNT= 177546  | ZF.SLI= 010000    |
| CE.SRC= 100364 | CS.RSN= 010000  | FC.PWR= 000022  | K\$CSR= 177546  | ZF.TIM= 000200    |
| CE.STP= 100352 | CS.SHU= 000001  | FC.RCE= 000002  | K\$SLDC= 000000 | ZF.X3P= 000000    |
| CE.TME= 100354 | CS.SID= 000002  | FC.RCP= 000014  | K\$STPS= 000074 | ZS.ASN= 100000    |
| CE.TMO= 100374 | CS.STR= 000004  | FC.TJM= 000010  | LD\$LP= 000000  | ZS.BSY= 140000    |
| CE.UNS= 100344 | CS.SUC= 000001  | FC.XCP= 000012  | L\$ASG= 000000  | Z.AVL 000014      |
| CF.CHN= 000001 | CS.TMO= 020000  | FC.XME= 000000  | L\$DRV= 000000  | Z.DAT 000016      |
| CF.DDM= 000002 | CS.XUR= 000004  | FS.AST= 000000  | L\$P11= 000001  | Z.DSP 000000      |
| CF.DYN= 000004 | C\$CKP= 000000  | FS.CJB= 002000  | L\$P11R= 000000 | Z.FLG 000010      |
| CF.EIS= 000010 | C\$ORE= 000400  | FS.CRA= 001000  | M\$CRB= 000124  | Z.LEN = 000016    |
| CF.EOM= 000004 | C\$SRSH= 177564 | FS.DIS= 013000  | M\$CRX= 000000  | Z.LLN 000006      |
| CF.FRK= 100000 | C.ADD 000034    | FS.DVC= 001000  | M\$FCFS= 000000 | Z.MAP 000020      |
| CF.HDR= 000020 | C.BID 000003    | FS.ENB= 012000  | M\$MGFE= 000000 | Z.NAM 000004      |
| CF.LB = 100000 | C.BUF 000014    | FS.EXI= 001000  | M\$NET= 000000  | Z.PCB 000012      |
| CF.LIN= 000002 | C.BUF1 000014   | F.GET= 006000   | M\$OVR= 000000  | Z.SCH 000007      |
| CF.LOG= 000020 | C.BUF2 000024   | FS.HLT= 000000  | N\$ACC= 000001  | \$CEPWR 000304RG  |
| CF.MDM= 000001 | C.CNT 000020    | FS.INI= 000000  | N\$BUF= 000001  | \$CMFRK= ***** GX |
| CF.SOM= 000010 | C.CNT1 000020   | FS.KIL= 000000  | N\$LDV= 000001  | \$CMPDV= ***** GX |
| CF.SYN= 000040 | C.CNT2 000030   | FS.LCL= 100000  | N\$MCP= 000001  | \$CXOPT= ***** GX |
| CF.TIM= 000400 | C.FLG 000022    | FS.LTM= 001000  | N\$MML= 000001  | \$DDFNC= ***** GX |
| CF.TRN= 000100 | C.FLG1 000022   | FS.MNT= 004000  | N\$MOV= 000010  | \$FRKHD= ***** GX |
| CM.CIR= 000002 | C.FLG2 000032   | FS.MSN= 014000  | N\$NCT= 000001  | \$INTSX 000000RG  |
| CM.FMT= 100000 | C.FNC 000010    | FS.REA= 001000  | N\$PEM= 000001  | \$INTX1= ***** GX |
| CM.HRD= 000002 | C.LIN 000006    | FS.RET= 000000  | PR7 = *****     | \$INTX7 000042RG  |
| CM.LIN= 000000 | C.LNK 000000    | FS.REZ= 003000  | PS = *****      | \$PDOSP 000076RG  |
| CM.LOD= 000001 | C.MOD 000011    | FS.RLB= 002000  | P\$P4S= 000000  | \$PDQUE 000170RG  |
| CM.XLO= 000004 | C.NSP 000004    | FS.RNG= 011000  | P\$WRD= 000000  | \$PDQU1 000174RG  |
| CP.DCF= 000040 | C.PRO 000042    | FS.RST= 000000  | Q\$SOP= 000010  | \$PDSPL 000052RG  |
| CP.HDL= 000007 | C.RSV 000002    | FS.RTN= 001000  | R\$SDE= 000000  | \$PDVTA ***** GX  |
| CP.PS = 177400 | C.STA 000007    | FS.SET= 005000  | R\$SK11= 000001 | \$STKDP= ***** GX |
| CP.PST= 000200 | C.STS 000012    | FS.SFC= 005000  | R\$SND= 000000  | .\$\$\$\$= 000034 |
| CP.XCF= 000100 | C.URM 177776    | FS.SFR= 006000  | R\$11M= 000000  |                   |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000320 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

```

205 .SBTTL $MVTBF - MOVE FROM MAPPED BUFFER TO UNMAPPED BUFFER
206
207
208 *** $MVTBF - MOVE FROM MAPPED BUFFER TO UNMAPPED BUFFER
209
210 THIS ROUTINE IS CALLED TO MOVE A BLOCK OF MEMORY (LESS THAN
211 4K WORDS) FROM A BUFFER THAT IS CURRENTLY MAPPED (EITHER
212 IN SYSTEM DYNAMIC SPACE OR IN THE COMM BUFFER POOL) TO AN
213 UNMAPPED BUFFER. NOTE THE MAPPED BUFFER CAN NOT BE IN THE PROCESS'S
214 SPACE MAPPED VIA APR5 SINCE IT USES APR5 TO MAP TO THE UNMAPPED
215 BUFFER.
216
217 CALLING FORMAT:
218 JSR R1,$MVTBF
219
220 INPUTS:
221 R2 = VIRTUAL ADDRESS OF MAPPED 'FROM' BUFFER
222 R3 = NUMBER OF BYTES TO MOVE
223 ON THE STACK:
224
225 R1 ORIGINAL CONTENTS OF R1 BEFORE CALL
226 VA 16-BIT VIRTUAL ADDRESS OF UNMAPPED 'TO' BFFER
227 BIAS RELOCATION BIAS OF UNMAPPED 'TO' BUFFER
228 (MAPPED SYSTEMS ONLY)
229
230
231 OUTPUTS:
232 R2 = UPDATED ADDRESS OF 'FROM' BUFFER
233 POINTS TO LAST BYTE MOVED +1
234 R3 = ZERO
235
236 STILL MAPPED TO 'FROM' BUFFER VIA KISAR6
237
238 REGISTERS MODIFIED:
239 R2 & R3
240
241
242
243 .ENABL LSB
244 $MVTBF::
245 000176 016746 000000G MOV KISAR5, -(SP) ; SAVE CURRENT PROCESS MAPPING
246 000202 016667 000006 000000G MOV 6(SP), KISAR5 ; MAP TO 'TO' BUFFER
247 000210 010166 000006 MOV R1, 6(SP) ; SAVE RETURN ADDRESS
248 000214 016601 000004 MOV 4(SP), R1 ; GET 'TO' BUFFER VIRTUAL ADDRESS
249 000220 022701 140000 CMP #140000, R1 ; IS THE 'TO' BUFFER IN THE EXEC POOL?
250 000224 101002 SUB #10, R1 ; IF HI, YES THEN DON'T ALTER THE VIRTUAL ADDRESS
251 000226 162701 020000 BHI 10$; SET VA FOR BIAS VIA KISAR5
252 000232 112221 MOVB (R2)+, (R1)+ ; MOVE BUFFER A BYTE AT A TIME
253 000234 SOB R3, 10$; LOOP TILL DONE
254 000240 000421 BR 20$; JOIN COMMON EXIT CODE

```

CESUB      CREATED BY    MACRO    ON 28-JUN-85 AT 18:20      PAGE 3      M 16

MACRO CROSS REFERENCE      CREF    04.00

MACRO NAME      REFERENCES

|         |        |        |        |        |        |        |        |
|---------|--------|--------|--------|--------|--------|--------|--------|
| CALL    | 10-355 |        |        |        |        |        |        |
| CALLR   | #4-62  | 13-477 | 14-511 |        |        |        |        |
| CCBDF\$ | #4-61  | 4-63   |        |        |        |        |        |
| CLKDF\$ | #4-61  | 4-65   |        |        |        |        |        |
| ENABL\$ | #4-60  |        |        |        |        |        |        |
| INHIB\$ | #4-60  |        |        |        |        |        |        |
| POVDF\$ | #4-61  | 4-64   |        |        |        |        |        |
| RESRG   | #4-60  | 5-98   | 6-139  | 10-353 | 11-412 | 12-446 |        |
| RETURN  | 5-99   | 6-140  | 7-202  | 9-306  | 10-363 | 11-413 | 12-447 |
| SAVRG   | #4-60  | 5-90   | 6-128  | 10-334 | 11-393 | 12-430 |        |
| SOB     | 8-253  | 9-302  |        |        |        |        |        |

```

342 .SBITL $RDBWT - Queue a request for a Receive Data Buffer
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358 000456 $RDBWT::SAVRG <R4> ; Save a register
359
360 .IF DF M$$PRO
361 CALL $MPSAV ; Bypass the cache
362 .ENDC
363
364 000460 006304 ASL R4 ; Convert SLN to word offset
365 000462 066704 000000G ADD $SLTMA,R4 ; Point into System Line Index Table
366 000466 105234 INCB @R4+ ; Increment wait request count for this line
367 000470 005267 000000G INC $RDQCT ; Increment global count
368
369
370 .IF DF M$$PRO
371 CALL @SP+ ; Restore state of the cache
372 .ENDC
373 000474 RESRG <R4> ; Restore the register
374 000476 RETURN

```

CEBUF

CREATED BY MACRO ON 28-JUN-85 AT 18:17

PAGE 3

N 2

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                   |
|---------|------------|------------------------------|
| \$RDBGI | 000044 RG  | 7-133 #8-156                 |
| \$RDBLH | = ***** GX | 8-172                        |
| \$RDBOP | 000500 RG  | #14-390                      |
| \$RDBRT | 000170 RG  | #9-210                       |
| \$RDBSZ | = ***** GX | 8-170 8-171                  |
| \$RDBTH | = ***** GX | 7-132                        |
| \$RDBWT | 000456 RG  | #13-358                      |
| \$RDQCT | = ***** GX | 9-216 *9-219 *13-367 *14-401 |
| \$SDBCT | = ***** GX | 12-320                       |
| \$SDBLH | = ***** GX | 11-295                       |
| \$SDBSZ | = ***** GX | 11-293 11-294                |
| \$SLTMA | = ***** GX | 13-365 14-397                |

```

342 .SBTTL $RDBWT - Queue a request for a Receive Data Buffer
343
344 :+
345 :*- $RDBWT- Queue a request for a Receive Data Buffer
346 :
347 : This routine is called by DDM processes after a failure of the
348 : DDM to allocate an Receive Data Buffer ($RDBGI) has occurred.
349 : The system line table flags word is updated to mark the buffer
350 : wait request and the global request is incremented.
351 :
352 : Inputs:
353 : R4 contains the System Line Number
354 :
355 : Outputs:
356 : None.
357 :-
358 000456 $RDBWT::SAVRG <R4> ; Save a register
359
360 .IF DF M$$PRO
361 CALL $MPSAV ; Bypass the cache
362 .ENDC
363
364 000460 006304 ASL R4 ; Convert SLN to word offset
365 000462 066704 ADD $SLTMA,R4 ; Point into System Line Index Table
366 000466 105234 INCB @(R4)+ ; Increment wait request count for this line
367 000470 005267 INC $RDQCT ; Increment global count
368
369 .IF DF M$$PRO
370 CALL @(SP)+ ; Restore state of the cache
371 .ENDC
372
373 000474 RESRG <R4> ; Restore the register
374 000476 RETURN

```

## SYMBOL CROSS REFERENCE

CPEF 04.00

| SYMBOL  | VALUE       | REFERENCES                     |
|---------|-------------|--------------------------------|
| \$CSBRT | = 000376 RG | #13-317                        |
| \$DEACB | = ***** GX  | 17-525                         |
| \$LDBAF | = ***** GX  | *8-134                         |
| \$LDBGI | = 000026 RG | #8-132                         |
| \$LDBRT | = 000170 RG | #10-209                        |
| \$PDQUE | = ***** GX  | 10-227                         |
| \$RDBCT | = ***** GX  | 8-132 10-215                   |
| \$RDBGI | = 000044 RG | 8-133 #9-156                   |
| \$RDBLH | = ***** GX  | 9-172                          |
| \$RDBNM | = ***** GX  | 16-444                         |
| \$RDBQP | = 000500 RG | #15-390                        |
| \$RDBRT | = 000170 RG | #10-210                        |
| \$RDBSZ | = ***** GX  | 9-170                          |
| \$RDBTH | = ***** GX  | 8-132 16-441 17-511            |
| \$RDBWT | = 000456 RG | #14-358                        |
| \$RDOCT | = ***** GX  | 10-216 *10-219 *14-367 *15-401 |
| \$SDBCT | = ***** GX  | 13-520                         |
| \$SDBLH | = ***** GX  | 12-295                         |
| \$SDBSZ | = ***** GX  | 12-294                         |
| \$SLTMA | = ***** GX  | 14-365 15-397                  |



CEDDM - MACRO V05.03b Friday 28-Jun-85 18:17 Page 10-1  
DDCM4 - COMMON PROCESS FOR FUNCTIONS WITH CCB

N 5

374

.DSABL LSB

\*\*FILF\*\*ID\*\*CEDMM

N 6

|          |            |          |          |     |     |     |     |
|----------|------------|----------|----------|-----|-----|-----|-----|
| CCCCCCCC | EEEEEEEEEE | DDDDDDDD | DDDDDDDD | MM  | MM  | MM  | MM  |
| CCCCCCCC | EEEEEEEEEE | DDDDDDDD | DDDDDDDD | MM  | MM  | MM  | MM  |
| CC       | EE         | DD       | DD       | MMM | MMM | MMM | MMM |
| CC       | EE         | DD       | DD       | MMM | MMM | MMM | MMM |
| CC       | EE         | DD       | DD       | MM  | MM  | MM  | MM  |
| CC       | EE         | DD       | DD       | MM  | MM  | MM  | MM  |
| CC       | EEEEEEEE   | DD       | DD       | MM  | MM  | MM  | MM  |
| CC       | EEEEEEEE   | DD       | DD       | MM  | MM  | MM  | MM  |
| CC       | EE         | DD       | DD       | MM  | MM  | MM  | MM  |
| CC       | EE         | DD       | DD       | MM  | MM  | MM  | MM  |
| CC       | EE         | DD       | DD       | MM  | MM  | MM  | MM  |
| CC       | EE         | DD       | DD       | MM  | MM  | MM  | MM  |
| CC       | EE         | DD       | DD       | MM  | MM  | MM  | MM  |
| CC       | EE         | DD       | DD       | MM  | MM  | MM  | MM  |
| CCCCCCCC | EEEEEEEEEE | DDDDDDDD | DDDDDDDD | MM  | MM  | MM  | MM  |
| CCCCCCCC | EEEEEEEEEE | DDDDDDDD | DDDDDDDD | MM  | MM  | MM  | MM  |

|            |          |            |
|------------|----------|------------|
| LL         | SSSSSSSS | TTTTTTTTTT |
| LL         | SSSSSSSS | TTTTTTTTTT |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SSSSSS   | TT         |
| LL         | SSSSSS   | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LLLLLLLLLL | SSSSSSSS | TT         |
| LLLLLLLLLL | SSSSSSSS | TT         |

```

424 .SBTTL $DDAST - ASYNCHRONOUS COMPLETION TO DLC LEVEL
425
426 ;+
427 ; **-$DDAST-ASYNCHRONOUS COMPLETION TO DATA LINK CONTROL
428 ;
429 ; THIS SUBROUTINE IS CALLED BY DEVICE DRIVERS WHEN AN ASYNCHRONOUS
430 ; CONDITION MUST BE RETURNED TO A DATA LINK CONTROL MODULE AND NO CCB
431 ; IS AVAILABLE AT THE DEVICE DRIVER.
432 ;
433 ; INPUTS:
434 ;
435 ; R3 = ASYNCHRONOUS COMPLETION STATUS (MOVED INTO C.STS)
436 ; R4 = SYSTEM LINE NUMBER
437 ;
438 ; OUTPUTS:
439 ;
440 ; A CCB IS ALLOCATED ON BEHALF OF THE DEVICE DRIVER
441 ; AND THE ASYNCHRONOUS STATUS IS QUEUED TO THE
442 ; DATA LINK CONTROL MODULE WITH:
443 ; C.FNC= FC.CCP
444 ; C.MOD= FS.AST
445 ;
446 ; REGISTERS MODIFIED:
447 ;
448 ; XXX
449 ;
450 ; .ENABL LSB
451 ;
452 000400 010446 $DDAST::MOV R4,-(SP) ; SAVE LINE NUMBER
453 000402 CALL $CCBGT ; ALLOCATE A CCB
454 000406 103406 BCS 5$; IF CS ERROR
455 000410 012664 000006 MOV (SP)+,C.LIN(R4) ; SET LINE NUMBER IN CCB
456 000414 012764 000020 000010 MOV #FC.CCP+FS.AST,C.FNC(R4) ; SET ERROR COMPLETE FUNCTION CODE
457 000422 000426 BR 20$; FINISH IN COMMON CODE
458
459 000424 012604 5$: MOV (SP)+,R4 ; CLEAN THE STACK
460 000426 RETURN ; RETURN TO CALLER
461

```

1 .IIF DF X\$\$NDM .TITLE CEDDMN  
2 .IIF DF X\$\$MDC .TITLE CEDDMN  
3 .IIF NDF X\$\$NDM & X\$\$MDC .TITLE CEDDM  
4 .IDENT /V05.00/  
5  
6  
7  
8  
9

10 : COPYRIGHT (C) 1978,1979,1980, 1982, 1983, 1985 BY  
11 : DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.  
12  
13

14 : THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
15 : ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
16 : INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
17 : COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
18 : OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
19 : TRANSFERRED.  
20

21 : THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
22 : AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
23 : CORPORATION.  
24

25 : DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
26 : SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
27

28 : MODULE DESCRIPTION:  
29

30 : CEX DDM/DLC INTERFACE ROUTINES  
31

32 : DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING  
33

34 : IDENT HISTORY:  
35

36 : 1.00 10-FEB-78  
37 : VERSION 2.0 RELEASE  
38

39 : 2.00 14-DEC-79  
40 : DECNET-11M/S V3.0  
41 : DECNET-11M-PLUS V1.0  
42

43 : 3.00 16-APR-82  
44 : DECNET-11M V3.1  
45 : DECNET-11M-PLUS V1.1  
46

47 : 4.00 07-NOV-83  
48 : DECNET-11M V4.0  
49 : DECNET-11M-PLUS V2.0  
50

51 : 5.00 22-JUL-85  
52 : DECnet-11M/S V4.2  
53 : DECnet-11M-Plus V3.0  
54 : DECnet-Micro/Rsx V1.0  
55  
56  
57

```
520 ADD $SLTMA,R3 ; POINT INTO SYSTEM LINE INDEX TABLE
521 MOV (R3),R3 ; GET ADDRESS OF SYSTEM LINE TABLE
522 MOV L.KRBA(R3),R3 ; GET POINTER TO KRB
523 MOV K.URM(R3),C.URM(R4)
524
525 .ENDC
526
527 CALLR $PDQU1 ; QUEUE CCB AND SCHEDULE PROCESS
528
529 .DSABL LSB
530
531 .ENDC
```

```

147 .SBTTL $XMCMP - TRANSMIT COMPLETE TO LLC LEVEL
148 .SBTTL $CTCMP - CONTROL COMPLETE TO LLC LEVEL
149 .SBTTL $RCCMP - RECEIVE COMPLETE TO LLC LEVEL
150 .SBTTL $KLCMP - KILL COMPLETE TO LLC LEVEL
151
152 +
153 ***$XMCMP-TRANSMIT COMPLETE TO LOGICAL LINK CONTROL PROCESSES
154 ***$CTCMP-CONTROL COMPLETE
155 ***$RCCMP-RECEIVE COMPLETE
156 ***$KLCMP-KILL COMPLETE
157
158 INPUTS:
159
160 R3 = OPERATION COMPLETION STATUS
161 R4 = ADDRESS OF CCB (OR FIRST CCB IN A CHAIN)
162 THE CCB CONTAINS A VALID LINE NUMBER
163
164 OUTPUTS:
165
166 THE APPROPRIATE FUNCTION CODE IS INSERTED INTO THE CCB,
167 THE LINE NUMBER IS USED TO FIND THE PDV SPECIFICATION, AND
168 THE CCB IS ADDED TO THE PROCESS QUEUE.
169
170 REGISTERS MODIFIED:
171
172 R3
173
174
175 000056 112764 000012 000010 $XMCMP:MOVB #FC.XCP,C.FNC(R4) ; SET TRANSMIT COMPLETE FUNCTION CODE
176 000064 000413 BR 10$; JOIN COMMON CODE
177
178 000066 112764 000020 000010 $CTCMP:MOVB #FC.CCP,C.FNC(R4) ; SET CONTROL COMPLETE FUNCTION CODE
179 000074 000407 BR 10$; JOIN COMMON CODE
180
181 000076 112764 000016 000010 $KLCMP:MOVB #FC.KCP,C.FNC(R4) ; SET KILL COMPLETE FUNCTION CODE
182 000104 000403 BR 10$; JOIN COMMON CODE
183
184 000106 112764 000014 000010 $RCCMP:MOVB #FC.RCP,C.FNC(R4) ; SET RECEIVE COMPLETE FUNCTION CODE
185 000114 010246 10$: MOV R2,-(SP) ; SAVE CALLERS R2
186
187 000116 116402 000006 MOVB C.LIN(R4),R2 ; GET SYSTEM LINE NUMBER
188
189 000122 010364 000012 20$: MOV R3,C.STS(R4) ; STORE OPERATION STATUS IN CCB
190 000126 142764 000200 000003 BICB #200,C.BID(R4) ; MARK MESSAGE FROM A DLC PROCESS
191 000134 006302 ASL R2 ; MULTIPLY SLN BY 2
192 000136 066702 000000G ADD $LLCTA,R2 ; POINT TO REVERSE MAPPING TABLE
193 000142 011203 MOV (R2),R3 ; GET SLN & STATION TO PDV & CHANNEL MAP ENTRY
194 000144 100005 BPL 25$; IF PL, GOT PDV & CHANNEL (POINT TO POINT)
195 ; ELSE, POINTER TO STATION TABLE MAP
196 000146 116402 000007 MOVB C.STA(R4),R2 ; GET STATION NUMBER (NEVER SIGN EXTENDS)
197 000152 060302 ADD R3,R2 ; COMPUTE HALF MAPPING TABLE ENTRY ADDRESS
198 000154 006302 ASL R2 ; MAKE A WORD ALIGNED ADDRESS
199 000156 011203 MOV (R2),R3 ; GET PDV & CHANNEL FOR THIS STATION
200 000160 052703 100000 25$: BIS #100000,R3 ; INDICATE THAT CELL CONTAINS A PDV & CHANNEL
201
202 .IF DF M$$PRO
203

```

CELLC MACRO V05.03b Friday 28-Jun-85 18:19 Page 5-1  
\$LLCRQ - LLC TO DLC REQUEST QUEUING SUBROUTINE

N 11

122 000052  
123 000056  
124 000072  
125  
126

MIPS #0 DROP PRIORITY TO 0 AND CLEAR CONDITION CODES  
RESRG <R5,R4,R3,R2,R1,R0> ; RESTORE REGISTERS  
REI IRN ; RETURN TO CALLING LLC  
; DF M\$\$PRO  
.ENDC

```

113
114 SAVRG <R4> ; PRESERVE CCB ADDRESS
115 MOV $LGDDB,R5 ; SET ADDRESS OF LOGGING DATABASE
116 MOV #SELFN,R3 ; SET UP DISPATCH FUNCTION
117 CALL $PDDSP ; AND DISPATCH TO LOGGING PROCESS
118 RESRG <R4> ; RESTORE CCB ADDRESS
119
120 CMP (SP)+,(SP)+ ; CLEAN UP THE STACK
121 TST (SP)+ ; ...
122
123 .ENDC
124
125 000000 10$: RETURN ; AND RETURN
126
127 000001 .END

```



CESCH MACRO V05.03b Friday 28-Jun-85 18:20  
Table of contents

|     |     |                                                              |
|-----|-----|--------------------------------------------------------------|
| 5-  | 74  | \$INTSX - COMM EXEC DEVICE INTERRUPT SAVE ROUTINE            |
| 6-  | 150 | \$INTX7 - LEVEL 7 INTERRUPT EXIT                             |
| 7-  | 180 | \$PDSPL - SET UP A PROCESS LEVEL AND DISPATCH TO IT          |
| 8-  | 217 | \$PDDSP - DISPATCH TO A PROCESS                              |
| 9-  | 283 | \$PDQUE - QUEUE A CCB (CHAIN) TO A LIST AND SCHEDULE PROCESS |
| 10- | 352 | POWERFAIL RECOVERY ROUTINE                                   |

CESCH MACRO V05.03b Friday 28-Jun-85 18:20 Page 10-2  
Symbol table

## \*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 15911 Words ( 63 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:13.48

SY:CESCH.V2,[130,134]CESCH/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]CESCH

CESUB MACRO V05.03b Friday 28-Jun-85 18:20 Page 9  
 \$MVFBF - MOVE FROM UNMAPPED BUFFER TO A MAPPED BUFFER

.SBTTL \$MVFBF - MOVE FROM UNMAPPED BUFFER TO A MAPPED BUFFER

\*\*\* \$MVFBF - MOVE FROM UNMAPPED BUFFER TO A MAPPED BUFFER

THIS ROUTINE IS CALLED TO MOVE A BLOCK OF MEMORY (LESS THAN  
 4K WORDS) FROM AN UNMAPPED BUFFER TO A BUFFER THAT IS CURRENTLY  
 MAPPED (EITHER IN SYSTEM DYNAMIC SPACE OR IN THE COMM BUFFER POOL).  
 THE ROUTINE UNMAPPS THE REQUESTING PROCESS AND USES ARP5 TO MAP  
 TO THE UNMAPPED "FROM" BUFFER.

CALLING FORMAT:

JSR R1,\$MVFBF

INPUTS:

R2 = VIRTUAL ADDRESS OF MAPPED "TO" BUFFER  
 R3 = NUMBER OF BYTES TO MOVE

ON STACK:

|      |                                                                    |
|------|--------------------------------------------------------------------|
| R1   | ORIGINAL CONTENTS OF R1 BEFORE CALL                                |
| VA   | 16-BIT VIRTUAL ADDRESS OF UNMAPPED "FROM" BUFFER                   |
| BIAS | RELOCATION BIAS OF UNMAPPED "FROM" BUFFER<br>(MAPPED SYSTEMS ONLY) |

OUTPUTS:

R2 = UPDATED ADDRESS OF "TO" BUFFER  
 POINTS TO LAST BYTE MOVED +1  
 R3 = ZERO

STILL MAPPED TO "TO" BUFFER VIA KISAR6

REGISTERS MODIFIED:

R2 & R3

\$MVFBF::

```

MOV * KISAR5,-(SP) ; SAVE CURRENT PROCESS MAPPING
MOV 6(SP),KISAR5 ; MAP TO "FROM" BUFFER
MOV R1,6(SP) ; SAVE RETURN ADDRESS
MOV 4(SP),R1 ; GET "FROM" BUFFER VIRTUAL ADDRESS
CMP #14000,R1 ; IS THE "FROM" BUFFER IN THE EXEC POOL?
BHI 15$; IF HI, YES THEN DON'T ALTER THE VIRTUAL ADDRESS
SUB #20000,R1 ; SET VA FOR BIAS VIA KISAR5
15$: MOV B (R1)+(R2)+ ; MOVE BUFFER A BYTE AT A TIME
 SOB R3,15$; LOOP TILL DONE
20$: MOV (SP)+,KISAR5 ; RESTORE PROCESS MAPPING
 MOV (SP)+,R1 ; RESTORE REGISTER
 TST (SP)+ ; CLEAN UP STACK
 RETURN ; RETURN

```

.DSABL LSB

256  
 257  
 258  
 259  
 260  
 261  
 262  
 263  
 264  
 265  
 266  
 267  
 268  
 269  
 270  
 271  
 272  
 273  
 274  
 275  
 276  
 277  
 278  
 279  
 280  
 281  
 282  
 283  
 284  
 285  
 286  
 287  
 288  
 289  
 290  
 291  
 292  
 293  
 294  
 295  
 296  
 297  
 298  
 299  
 300  
 301  
 302  
 303  
 304  
 305  
 306  
 307  
 308

```

000242 016746 000000G 000000G
000242 016667 000006 000000G
000254 010166 000006
000260 016601 000004
000264 022701 1'0000
000270 101002
000272 162701 020000
000276 112122
000300
000304 012667 000000G
000310 012601
000312 005726
000314

```

|   |      |
|---|------|
| 1 | CEBU |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |
| G |      |
| H |      |
| I |      |
| J |      |
| K |      |
| L |      |
| M |      |
| N |      |
| B |      |
| C |      |
| D |      |
| E |      |
| F |      |

|   |    |        |
|---|----|--------|
| J | 5  |        |
| K | 5  |        |
| L | 5  |        |
| M | 5  |        |
| N | 5  |        |
| B | 6  |        |
| C | 6  |        |
| D | 6  |        |
| E | 6  |        |
| F | 6  |        |
| G | 6  |        |
| H | 6  |        |
| I | 6  |        |
| J | 6  |        |
| K | 6  |        |
| L | 6  |        |
| M | 6  |        |
| N | 6  |        |
| B | 7  | CEDDMM |
| C | 7  |        |
| D | 7  |        |
| E | 7  |        |
| F | 7  |        |
| G | 7  |        |
| H | 7  |        |
| I | 7  |        |
| J | 7  |        |
| K | 7  |        |
| L | 7  |        |
| M | 7  |        |
| N | 7  |        |
| B | 8  |        |
| C | 8  |        |
| D | 8  |        |
| E | 8  |        |
| F | 8  |        |
| G | 8  |        |
| H | 8  |        |
| I | 8  |        |
| J | 8  |        |
| K | 8  |        |
| L | 8  |        |
| M | 8  |        |
| N | 8  |        |
| B | 8  | CEDDMM |
| C | 8  |        |
| D | 8  |        |
| E | 8  |        |
| F | 8  |        |
| G | 8  |        |
| H | 8  |        |
| I | 8  |        |
| J | 8  |        |
| K | 8  |        |
| L | 8  |        |
| M | 8  |        |
| N | 8  |        |
| B | 9  |        |
| C | 9  |        |
| D | 9  |        |
| E | 9  |        |
| F | 9  |        |
| G | 9  |        |
| H | 9  |        |
| I | 9  |        |
| J | 9  |        |
| K | 9  |        |
| L | 9  |        |
| M | 9  |        |
| N | 9  |        |
| B | 10 |        |
| C | 10 |        |
| D | 10 |        |

|   |    |       |
|---|----|-------|
| E | 10 |       |
| F | 10 |       |
| G | 10 | CECLC |
| H | 10 |       |
| I | 10 |       |
| J | 10 |       |
| K | 10 |       |
| L | 10 |       |
| M | 10 |       |
| N | 11 |       |
| B | 11 |       |
| C | 11 |       |
| D | 11 |       |
| E | 11 |       |
| F | 11 |       |
| G | 11 | CELLC |
| H | 11 |       |
| I | 11 |       |
| J | 11 |       |
| K | 11 |       |
| L | 11 |       |
| M | 11 |       |
| N | 12 |       |
| B | 12 |       |
| C | 12 |       |
| D | 12 |       |
| E | 12 |       |
| F | 12 |       |
| G | 12 |       |
| H | 12 | CELOG |
| I | 12 |       |
| J | 12 |       |
| K | 12 |       |
| L | 12 |       |
| M | 13 |       |
| N | 13 |       |
| B | 13 |       |
| C | 13 | CELOG |
| D | 13 |       |
| E | 13 |       |
| F | 13 |       |
| G | 13 |       |
| H | 13 |       |
| I | 13 |       |
| J | 13 | CESCH |
| K | 13 |       |
| L | 13 |       |
| M | 13 |       |
| N | 14 |       |
| B | 14 |       |
| C | 14 |       |
| D | 14 |       |
| E | 14 |       |
| F | 14 |       |
| G | 14 |       |
| H | 14 |       |
| I | 14 |       |
| J | 14 |       |
| K | 14 |       |
| L | 14 |       |

|   | 14 | 15 | 16 |
|---|----|----|----|
| M | 14 | 15 | 16 |
| N | 14 | 15 | 16 |
| B | 14 | 15 | 16 |
| C | 14 | 15 | 16 |
| D | 14 | 15 | 16 |
| E | 14 | 15 | 16 |
| F | 14 | 15 | 16 |
| G | 14 | 15 | 16 |
| H | 14 | 15 | 16 |
| I | 14 | 15 | 16 |
| J | 14 | 15 | 16 |
| K | 14 | 15 | 16 |
| L | 14 | 15 | 16 |
| M | 14 | 15 | 16 |
| N | 14 | 15 | 16 |
| B | 14 | 15 | 16 |
| C | 14 | 15 | 16 |
| D | 14 | 15 | 16 |
| E | 14 | 15 | 16 |
| F | 14 | 15 | 16 |
| G | 14 | 15 | 16 |
| H | 14 | 15 | 16 |
| I | 14 | 15 | 16 |
| J | 14 | 15 | 16 |
| K | 14 | 15 | 16 |
| L | 14 | 15 | 16 |
| M | 14 | 15 | 16 |

\*\*\*FILE\*\*ID\*\*CESUB1

B 1

```
CCCCCCCC EEEEEEEEE SSSSSSSS UU UU BBBB8888 11
CCCCCCCC EEEEEEEEE SSSSSSSS UU UU BBBB8888 11
CC EE SS UU UU BB BB 1111
CC EE SS UU UU BB BB 1111
CC EE SS UU UU BB BB 11
CC EE SS UU UU BB BB 11
CC EEEEEEE SSSSSS UU UU BBBB8888 11
CC EEEEEEE SSSSSS UU UU BBBB8888 11
CC EE SS UU UU BB BB 11
CC EE SS UU UU BB BB 11
CC EE SS UU UU BB BB 11
CC EE SS UU UU BB BB 11
CCCCCCCC EEEEEEEEE SSSSSSSS UUUUUUUUU BBBB8888 111111
CCCCCCCC EEEEEEEEE SSSSSSSS UUUUUUUUU BBBB8888 111111
....
....
....
```

```
LL SSSSSSSS TTTTTTTTT
LL SSSSSSSS TTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
```

C 1

450  
 451  
 452  
 453  
 454  
 455  
 456  
 457  
 458  
 459  
 460  
 461  
 462  
 463  
 464  
 465  
 466  
 467  
 468  
 469  
 470  
 471  
 472  
 473  
 474  
 475  
 476  
 477  
 478  
 479

000536  
 000536 070:00  
 000540

```
.SBTTL $CEMUL - UNSIGNED MULTIPLY
;
; *- $CEMUL - UNSIGNED MULTIPLY
;
; THIS SUBROUTINE PERFORMS UNSIGNED MULTIPLICATION WHERE THE RESULT IS LESS
; THAN 16 BITS. IT IS INCLUDED HERE SO THAT WE CAN CONDITIONALISE THE CODE
; FOR EXTENDED INSTRUCTION SET.
;
; NOTE HAT IT IS THE CALLERS RESPONSIBILITY TO ENSURE THAT THE RESULT
; WILL FIT IN 16 BITS.
;
; INPUTS:
; R0 = MULTIPLIER
; R1 = MULTIPLICAND
;
; OUTPUTS:
; R0 = CORRUPTED
; R1 = LOW 16 BITS OF RESULT
;
$CEMUL::IF DF R$$EIS
 MUL R0,R1 ; PERFORM MULTIPLICATION
 RETURN
 .IFF ; DF R$$EIS
 CALLR $MUL ; CALL EXEC MULTIPLY ROUTINE
 .ENDC ; DF R$$EIS
```

```

61 .SBTTL TIMEOUT PROCESSOR
62
63 ;+
64 ***-$TSTIM-TIMEOUT PROCESSOR
65
66 THIS SUBROUTINE IS CALLED TO TEST AND DECREMENT A TIMEOUT COUNT IN
67 THE FIRST BYTE OF A PROCESS LINE TABLE. IF THE COUNT IS DECREMENTED
68 TO ZERO, THE PROCESS IS DISPATCHED AT IT'S TIMEOUT ENTRY POINT
69 WITH A 'FS.LTM' SUBFUNCTION CODE.
70
71 -
72 INPUTS:
73 R2 - PDV INDEX OF PROCESS
74 R5 - POINTER TO PROCESS LINE TABLE
75
76 OUTPUTS:
77 ONLY R4 WILL BE PRESERVED
78
79 $TSTIM::IF DF M$$MGE
80
81 000000 010203 MOV R2,R3 ; COPY PDV INDEX
82 000002 066703 ADD $PDVTA,R3 ; POINT INTO PDV INDEX TABLE
83 000006 016746 000000G MOV KISAR5,-(SP) ; SAVE CURRENT MAPPING
84 000012 013367 000000G MOV @ (R3)+,KISAR5 ; MAP TO PROCESS (FOR MAPPED LINE TABLES)
85
86 .ENDC
87
88 000016 MTPS #PR7 ;;; DISABLE INTERRUPTS
89 000024 TSTB (R5) ;;; IS THE TIMER ACTIVE?
90 000026 BEQ 10$;;; NO ... JUST EXIT
91 000030 DECB (R5) ;;; YES ... DECREMENT TIME TO GO
92 000032 BNE 10$;;; IF NE, TIMER STILL RUNNING
93 000034 CALL $DSPTM ;;; DISPATCH TO PROCESS
94 000040 10$: MTPS #0 ;;; RESTORE PRIORITY
95
96 .IF DF M$$MGE
97
98 000044 012667 000000G MOV (SP)+,KISAR5 ;;; RESTORE MAPPING
99
100 .ENDC
101 RETURN
102

```

```

237 .WORD 0 ; TIMER QUEUE BLOCK
238 .WORD 0
239 .WORD 0
240 .ENDR
241
242 $$$OFF=2
243 $TSCTB:::REPT M$$PRO
244 .WORD TIMERS+$$$OFF ; POINTER TO PROCESSOR DEPENDANT TIMER CELL
245 $$$OFF=$$$OFF+$$$LEN
246 .ENDR
247
248 $$$OFF=$$$STM
249 $STMTB:::REPT M$$PRO
250 .WORD TIMERS+$$$OFF ; POINTER TO PROCESSOR DEPENDANT TIMER CELL
251 $$$OFF=$$$OFF+$$$LEN
252 .ENDR
253
254 $CRESL::LCKDF$ SPIN ; SERIALIZE ACCESS TO COMMEEXEC RESOURCES
255
256 .ENDC ; M$$PRO
257 .ENDC ; R$$MPL
258
259 000312 000000 $SHLST:::WORD 0 ; SECONDARY HOST LISTHEAD
260
261 000314 $IMASK:: ; For 3271 back compatibility (1.0 comm exec)
262 000314 $EMASK:: ; ...
263 000314 $ETIMR:: ; ...
264 000314 000000 $SPAR1:::WORD 0 ; First Double word spare cell
265 000316 000000 .WORD 0
266 000320 $PBIAS:: ; For 3271 back compatibility (1.0 comm exec)
267 000320 000000 $SPAR2:::WORD 0 ; Second Double word spare cell
268 000322 000000 .WORD 0
269
270 ; .IF DF R$$MPI ; Leave cells for M+ backwards compatability
271 ;
272 ; Cells to be moved to the DECnet Home Block
273 ;
274 $ANNHD:::BLKW 1 ; ALIAS NODE NAME LISTHEAD
275 $RNNHD:::WORD 1+2 ; REMOTE NODE NAME LISTHEAD
276 .WORD 0 ; TEMPORARY END OF REMOTE NAME LISTHEAD
277 $NTNAM:::BLKB 6 ; LOCAL NODE NAME
278 $NSPNM:::BLKW 1 ; LOCAL NODE NUMBER
279 $NODID:::BLKW 1 ; LENGTH OF LOCAL NODE IDENT STRING
280 .BLKB 32. ; LOCAL NODE IDENTIFICATION
281 $HOST:::BLKW 1 ; HOST NODE ADDRESS
282 $NLN:::BLKW 1 ; NUMBER OF ROUTING CHANNELS FOR NSP
283 $NN:::BLKW 1 ; NUMBER OF NODES IN NETWORK
284 $MAXC:::BLKW 1 ; MAXIMUM COST
285 $MAXH:::BLKW 1 ; MAXIMUM HOPS
286 $MAXV:::WORD 20. ; MAXIMUM VISITATION COUNT
287 $SQRTL:::BLKW 1 ; SQUARE ROOT LIMITING FACTOR
288 $ROUTM:::WORD 30. ; ROUTING TIMER (SECONDS)
289 ;
290 .ENDC ; R$$MPL
291 ;
292 ; SYNC AND PAD BUFFER DESCRIPTORS
293 ;

```



101

102

103 000000 000000

104 000002 000000G

105 000004 172354

106

107

000002

.SBTTL EXECUTIVE VECTOR TABLE

\$STCVT::WORD 0 ; FLAGS WORD

KSAR5: .WORD KISAR5 ;

KSARG: .WORD KIS'16 ;

\$STCVL == <<.-\$STCVT>/2>-1 ; LENGTH OF VECTOR TABLE

STCRC MACRO V05.03b Friday 28-Jun-85 18:21 Page 11-5  
MODIFIER TABLE FOR SOFTWARE CRC

B 6

716  
717  
718  
719

000001

.ENDC

.END

```

109 .SBTTL $CLCRC - CALCULATE CRC-16 ON A TRANSMIT CHAIN
110
111 *+-$CLCRC-CALCULATE CRC (CRC16) ON A TRANSMIT CHAIN
112
113 -
114 INPUTS:
115 R4 - ADDRESS OF FIRST CCB IN CHAIN
116 C.LNK - ADDRESS OF NEXT CCB IN CHAIN (0 MARKS END)
117 C.BUF - DOUBLE WORD ADDRESS OF BUFFER
118 C.CNT - LENGTH OF BUFFER
119 C.FLG - BUFFER FLAGS
120
121 OUTPUTS:
122 CRC IS STORED AT THE END OF EACH BUFFER AND THE BYTE
123 COUNT IS UPDATED.
124
125 REGISTERS MODIFIED:
126 R2, R3
127
128 $CLCRC::
129 .IF DF M$$MGE
130 .IF NDF I$$AS
131
132 000006 MOV @KSAR5,-(SP) ; SAVE CALLER'S MAPPING
133
134 .IFF ; I$$AS
135
136 MOV #KP.AR3,-(SP) ; ASSUME KERNEL MODE
137 BIT #140000,PS.EXP ; IS IT ?
138 BEQ 5$; YES - BR
139 MOV #UPAR0+6,(SP) ; NO - MUST BE USER MODE
140 5$: MOV @ (SP),-(SP) ; SAVE CURRENT MAPPING
141
142 .ENDC ; I$$AS
143 .ENDC ; M$$MGE
144
145 000012 SAVRG <R0,R1,R4> ; SAVE REGISTERS
146 000020 005001 CLR R1 ; INITIALISE CRC ACCUMULATOR
147
148 000022 010403 10$: MOV R4,R3 ; COPY CCB ADDRESS
149 000024 062703 000014 ADD #C.BUF,R3 ; AND POINT TO BUFFER DESCRIPTOR
150
151 .IF DF M$$MGE
152 .IF NDF I$$AS
153
154 000030 012377 177746 MOV (R3)+,@KSAR5 ; MAP TO THE DATA
155
156 .IFF ; I$$AS
157
158 MOV (R3)+,@10(SP) ; MAP TO THE DATA
159
160 .ENDC ; I$$AS
161 .IFF ; M$$MGE
162
163 TST (R3)+ ; SKIP OVER RELOCATION BIAS
164
165 .IFTF ; M$$MGE

```

STCRCF MACRO V05.03b Friday 28-Jun-85 18:22 Page 12-6  
Symbol table

|                |                  |                  |                  |                    |
|----------------|------------------|------------------|------------------|--------------------|
| ASSCHK= 000000 | CS.ABO= 000100   | C.STS 000012     | FS.RNG= 011000   | M\$\$NET= 000000   |
| ASSCPS= 000000 | CS.BRO= 000002   | C.URM 177776     | FS.RST= 000000   | M\$\$OVR= 000000   |
| ASSPRI= 000000 | CS.BUF= 000200   | C.XACP 000004    | FS.RTN= 001000   | N\$\$ACC= 000001   |
| ASSTRP= 000000 | CS.CES= 000002   | C.XID 000035     | FS.SET= 005000   | N\$\$BUF= 000001   |
| BUFUMP= 172354 | CS.CHN= 000010   | C.XLEN 000044    | FS.SFC= 005000   | N\$\$LDV= 000001   |
| CB.CCB= 000002 | CS.CMP= 000200   | C.XPLI 000040    | FS.SFR= 006000   | N\$\$MCP= 000001   |
| CB.DDM= 000040 | CS.DCR= 000400   | C.XPT 000034     | FS.SFS= 004000   | N\$\$MLL= 000001   |
| CB.DLC= 000020 | CS.DEF= 000004   | C.XSVC 000042    | FS.SPW= 040000   | N\$\$MOV= 000010   |
| CB.RDB= 000004 | CS.DEV= 000002   | C.XTC 000037     | FS.STM= 000000   | N\$\$NCT= 000001   |
| CB.SDB= 000010 | CS.DIS= 000040   | C.X25 000036     | FS.STP= 002000   | N\$\$PEM= 000001   |
| CB.SLI= 000100 | CS.ENA= 000001   | DDB = 000010     | FS.STR= 001000   | PIRQ = 177772      |
| CB.XLB= 000001 | CS.ENB= 000020   | D\$\$BUG= 177514 | FS.TRM= 003000   | PMODE = 030000     |
| CC.LLC= 000200 | CS.ERR= 100000   | D\$\$ISK= 000000 | FS.WLB= 001000   | PRO = 000000       |
| CE.ABO= 100362 | CS.FTL= 001000   | D\$\$L11= 000001 | FS.XKL= 002000   | PR1 = 000040       |
| CE.DAO= 100346 | CS.HCR= 000001   | D\$\$SYN= 000000 | FS.XOF= 010000   | PR2 = 000100       |
| CE.DIS= 100366 | CS.HFE= 002000   | D\$\$SYM= 000000 | FS.XON= 007000   | PR3 = 000140       |
| CE.ERP= 100370 | CS.LST= 040000   | E\$\$XPR= 000000 | FS.ZER= 002000   | PR4 = 000200       |
| CE.ILN= 100350 | CS.MTL= 004000   | FC.CCP= 000020   | F\$\$AST= 000001 | PR5 = 000240       |
| CE.LTO= 100356 | CS.RNG= 000010   | FC.CTL= 000006   | F\$\$LVL= 000001 | PR6 = 000300       |
| CE.MOP= 100372 | CS.ROV= 000004   | FC.KCP= 000016   | G\$\$TPP= 0000J0 | PR7 = 000340       |
| CE.NTE= 100361 | CS.RSN= 010000   | FC.KIL= 000004   | G\$\$TSS= 000000 | PS = 177776        |
| CE.RTE= 100376 | CS.SHU= 000001   | FC.MAN= 000024   | G\$\$TTK= 000000 | P\$\$P45= 000000   |
| CE.SRC= 100364 | CS.SID= 000002   | FC.MLD= 000026   | G\$\$WRD= 000000 | P\$\$WRD= 000000   |
| CE.STP= 100352 | CS.STR= 000004   | FC.PCT= 000030   | I\$\$RAR= 000000 | Q\$\$OPT= 000010   |
| CE.TME= 100354 | CS.SUC= 000001   | FC.PWR= 000022   | I\$\$RDN= 000000 | R\$\$DER= 000000   |
| CE.TMO= 100374 | CS.TMO= 020000   | FC.RCE= 000002   | KGCSR = 170700   | R\$\$EJS= 000000   |
| CE.UNS= 100344 | CS.XUR= 000004   | FC.RCP= 000014   | KGINIT= 000111   | R\$\$K11= 000001   |
| CF.CHN= 000001 | CTABL 000234P    | FC.TIM= 000010   | KGLDBC= 000133   | R\$\$SND= 000000   |
| CF.EOM= 000004 | C\$\$CKP= 000000 | FC.XCP= 000012   | KISAR0= 172340   | R\$\$11M= 000000   |
| CF.HDR= 000020 | C\$\$ORE= 000400 | FC.XME= 000000   | KISAR5= *****    | SEN = 000100       |
| CF.LE = 100000 | C\$\$RSR= 177564 | FS.AST= 000000   | KISAR6= 172354   | STCR2 000162R      |
| CF.LIN= 000002 | C.ADD 000034     | FS.CIB= 002000   | KSAR5 000002R    | SWR = 177570       |
| CF.SOM= 000010 | C.BID 000003     | FS.CRA= 001000   | KSAR6 000004R    | S\$\$WRG= 000000   |
| CF.SYN= 000040 | C.BUF 000014     | FS.DIS= 013000   | K\$\$CNT= 177546 | S\$\$YSZ= 007600   |
| CF.TRN= 000100 | C.BUF1 000014    | FS.DVC= 001000   | K\$\$CSR= 177546 | TPS = 177564       |
| CLRGK = 000020 | C.BUF2 000024    | FS.ENB= 012000   | K\$\$LDC= 000000 | T\$\$KMG= 000000   |
| CMODE = 140000 | C.CNT 000020     | FS.EXI= 001000   | K\$\$TPS= 000074 | T\$\$MIN= 000000   |
| CM.CIR= 000002 | C.CNT1 000020    | FS.HLT= 000000   | LD\$LP = 000000  | UBMPR = 170200     |
| CM.FMT= 100000 | C.CNT2 000030    | FS.INI= 000000   | LRC16 = 000003   | UISAR0= 177640     |
| CM.HRD= 000002 | C.FLC 000022     | FS.KIL= 000000   | L\$\$ASG= 000000 | UISAR1= 177642     |
| CM.LIN= 000000 | C.FLG1 000022    | FS.LCL= 100000   | L\$\$DRV= 000000 | V\$\$CTR= 001000   |
| CM.LOO= 000001 | C.FI G2 000032   | FS.LTM= 001000   | L\$\$P11= 000001 | X\$\$DBT= 000000   |
| CM.XLO= 000004 | C.FNC 000010     | FS.MNT= 004000   | L\$\$11R= 000000 | \$CLCRC 000006RG   |
| CP.DCF= 000040 | C.LIN 000006     | FS.MSN= 014000   | MPAR = 172100    | \$STCRC 000122RG   |
| CP.HDL= 000007 | C.LNK 000000     | FS.REA= 001000   | MPCSR = 177746   | \$STCR1 000124RG   |
| CP.PS = 177400 | C.MOD 000011     | FS.RET= 000000   | M\$\$CRB= 000124 | \$STCVL= 000002 G  |
| CP.PSI= 000200 | C.NSP 000004     | FS.REZ= 003000   | M\$\$CRX= 000000 | \$STCVT 000000RG   |
| CP.XCF= 000100 | C.PRO 000042     | FS.RLB= 002000   | M\$\$FCS= 000000 | .BASEB= 14.000     |
| CP.2FR= 000030 | C.RSV 000002     |                  | M\$\$MGE= 000000 | .\$\$\$\$.= 000034 |
| CRC16 = 000001 | C.STA 000007     |                  |                  |                    |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
001234 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

```

166
167 000034 012302 MOV (R3)+,R2 ; GET VIRTUAL ADDRESS OF BUFFER
168
169 .IFT ; M$$MGE
170 .IF NDF 1$$AS
171
172 000036 022702 140000 CMP #140000,R2 ; IS BUFFER MAPPED BY AND APR < 6?
173 000042 101002 BHI 20$; IF HI, YES
174 000044 162702 020000 SUB #20000,R2 ; MAP BUFFER THROUGH APR5
175
176 .IFF ; 1$$AS
177 CMP #60000,R2 ; IS BUFFER MAPPED BY AN APR 3 ?
178 BIOS 20$; IF LO OR SAME, NO - MUST BE APR 3
179 ADD #20000,R2 ; MAP BUFFER THROUGH APR3
180 .ENDC ; 1$$AS
181 000050 20$: .ENDC ; M$$MGE
182
183
184 000050 012300 MOV (R3)+,R0 ; GET LENGTH OF SEGMENT
185 000052 CALL STCR2 ; COMPUTE CRC ON SEGMENT
186 000056 032713 000004 BIT #CF.EOM,(R3) ; LAST SEGMENT IN MESSAGE?
187 000062 001407 BEQ 30$; IF EQ, NO
188
189 000064 060002 ADD R0,R2 ; POINT TO END OF MESSAGE
190 000066 110122 MOVB R1,(R2)+ ; STORE CRC FOLLOWING DATA
191 000070 000301 SWAB R1 ; ...
192 000072 110122 MOVB R1,(R2)+ ; ...
193 000074 062743 000002 ADD #2,-(R3) ; UPDATE COUNT IN CCB
194 000100 005001 CLR R1 ; RE-INITIALISE CRC ACCUMULATOR
195
196 000102 011404 30$: MOV (R4),R4 ; GET NEXT CCB IN CHAIN
197 000104 001346 BNE 10$; AND COMPUTE IT'S CRC
198
199 000106 RESRG <R4,R1,R0> ; RESTORE REGISTERS
200
201 .IF DF M$$MGE
202 .IF NDF 1$$AS
203
204 000114 012677 177662 MOV (SP)+,@KSAR5 ; RESTORE MAPPING
205
206 .IFF ; 1$$AS
207
208 MOV (SP)+,@(SP)+ ; RESTORE MAPPING
209
210 .ENDC ; 1$$AS
211 .ENDC ; M$$MGE
212
213 000120 RETURN

```

\*\*wjd01\*\*

STCRCK MACRO V05.03b Friday 28-Jun-85 18:22 Page 12-7  
Symbol table

B 10

Work file reads: 0  
Work file writes: 0  
Size of work file: 15190 Words ( 60 Pages)  
Size of core pool: 16552 Words ( 63 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:15.20

SY:STCRCK.V2,[130,134]STCRCK/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]V2,STCRCK

AXBFR MACRO V05.03b Friday 28-Jun-85 18:28 Page 6-4  
Symbol table

|                  |                |                  |                  |                    |
|------------------|----------------|------------------|------------------|--------------------|
| T\$\$KMG= 000000 | UBMPR = 170200 | UISAR1= 177642   | X\$\$DBT= 000000 | .BASEB= 140000     |
| T\$\$MIN= 000000 | UISAR0= 177640 | V\$\$CTR= 001000 | \$BFRIN 000000RG | .\$\$\$\$.= 000034 |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000214 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 3  
Work file writes: 4  
Size of work file: 17108 words ( 67 Pages)  
Size of core pool: 17608 words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:11.59  
SY:AXBFR.V2,[131,134]AXBFR/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,1]JAXBFR

AXDAT      CREATED BY    MACRO    ON 28-JUN-85 AT 18:28

PAGE 3      B 12

SYMBOL CROSS REFERENCE

CREF    04.00

| SYMBOL | VALUE | REFERENCES |
|--------|-------|------------|
|--------|-------|------------|

|         |                  |       |
|---------|------------------|-------|
| \$STMFC | =    *****    GX | 5-132 |
| \$TKPS  | =    *****    GX | 5-74  |
| \$TK100 | =    *****    GX | 5-138 |
| \$TSKRT | =    *****    GX | 5-75  |
| \$TSTIM | =    *****    GX | 5-139 |
| \$TISCL | =    *****    GX | 5-140 |
| \$T100C | =    *****    GX | 5-141 |
| \$T100Q | =    *****    GX | 5-142 |
| \$UMRPT | =    *****    GX | 5-76  |
| \$XAVL  | =    *****    GX | 5-143 |
| \$ZTIME | =    *****    GX | 5-144 |
| \$ZTIM2 | =    *****    GX | 5-148 |



|                 |                |                 |                |                   |
|-----------------|----------------|-----------------|----------------|-------------------|
| ASSCHK= 000000  | CS.DEV= 000002 | D\$SYNC= 000000 | G\$TSS= 000000 | L.NMST 000020     |
| ASSCPS= 000000  | CS.DIS= 000040 | D\$SYNM= 000000 | G\$TTK= 000000 | L.NSTA 000014     |
| ASSPRI= 000000  | CS.ENA= 000001 | EXRQN= ***** GX | G\$WRD= 000000 | L.OWNR 000021     |
| ASSTRP= 000000  | CS.ENB= 000020 | E\$XPR= 000000  | I\$RAR= 000000 | L.UNT 000013      |
| CB.CCB= 000002  | CS.ERR= 000000 | FC.CCP= 000020  | I\$RDN= 000000 | M\$SRB= 000124    |
| CB.DDM= 000040  | CS.FTL= 001000 | FC.CTL= 000006  | K\$CNT= 177546 | M\$SRX= 000000    |
| CB.DLC= 000020  | CS.HCR= 000001 | FC.KCP= 000016  | K\$CSR= 177546 | M\$FCS= 000000    |
| CB.RDB= 000004  | CS.HFE= 002000 | FC.KIL= 000004  | K\$LDC= 000000 | M\$MGE= 000000    |
| CB.SDB= 000010  | CS.LST= 040000 | FC.MAN= 000024  | K\$TPS= 000074 | M\$NET= 000000    |
| CB.SLI= 000100  | CS.MTL= 004000 | FC.MLD= 000026  | LD\$LP= 000000 | M\$OVR= 000000    |
| CB.XLB= 000001  | CS.RNG= 000010 | FC.PCT= 000030  | LF.ACT= 100000 | NETACP 000034R    |
| CCBRET= 000030R | CS.ROV= 000004 | FC.PWR= 000022  | LF.BRO= 000400 | NMCL2= ***** GX   |
| CCBRT= ***** GX | CS.RSN= 010000 | FC.RCE= 000002  | LF.BWT= 000007 | NMCMF 000040R     |
| CC.LLC= 000200  | CS.SHU= 000001 | FC.RCP= 000014  | LF.ENA= 002000 | N\$ACC= 000001    |
| CE.ABO= 100362  | CS.SID= 000002 | FC.TIM= 000010  | LF.LPB= 001000 | N\$BUF= 000001    |
| CE.DAO= 100346  | CS.STR= 000004 | FC.XCP= 000012  | LF.MDC= 000100 | N\$LDY= 000001    |
| CE.DIS= 100366  | CS.SUC= 000001 | FC.XME= 000000  | LF.MFL= 004000 | N\$MCP= 000001    |
| CE.ERR= 100370  | CS.TMO= 020000 | FS.AST= 000000  | LF.MTP= 000020 | N\$MILL= 000001   |
| CE.ILN= 100350  | CS.XUR= 000004 | FS.CIB= 002000  | LF.PAC= 000200 | N\$MOV= 000010    |
| CE.LTO= 100356  | D\$CKP= 000000 | FS.CRA= 001000  | LF.RDY= 040000 | N\$NCT= 000001    |
| CE.MOP= 100372  | D\$ORE= 000400 | FS.DIS= 013000  | LF.REA= 010000 | N\$PEM= 000001    |
| CE.NTE= 100361  | D\$RSH= 177564 | FS.DVC= 001000  | LF.SER= 000040 | P\$PG5= 000000    |
| CE.RTE= 100376  | C.ADD 000034   | FS.ENB= 012000  | LF.TIM= 000010 | P\$WRD= 000000    |
| CE.SRC= 100364  | C.BID 000003   | FS.EJI= 001000  | LF.UNL= 020000 | Q\$OPT= 000010    |
| CE.STP= 100352  | C.BUF 000014   | FS.GET= 006000  | LF.X2P= 000000 | RDBRT= ***** GX   |
| CE.TME= 100354  | C.BUF1 000014  | FS.HLT= 000000  | LN.CLO= 000000 | R\$DER= 000000    |
| CE.TMO= 100374  | C.BUF2 000024  | FS.INJ= 000000  | LN.DUM= 000005 | R\$K11= 000001    |
| CE.UNS= 100344  | C.CNT 000020   | FS.KIL= 000000  | LN.LOA= 000004 | R\$SND= 000000    |
| CF.CHN= 000001  | C.CNT1 000020  | FS.LCL= 100000  | LN.LOO= 000003 | R\$11M= 000000    |
| CF.EOM= 000004  | C.CNT2 000030  | FS.LTM= 001000  | LN.OAU= 000003 | SF.ACT= 000200    |
| CF.HDR= 000020  | C.FLG 000022   | FS.MNT= 004000  | LN.OFF= 000001 | SF.ENA= 000100    |
| CF.LB= 100000   | C.FLG1 000022  | FS.MSN= 014000  | LN.ON= 000000  | SF.LPB= 000004    |
| CF.LTI= 000002  | C.FLG2 000032  | FS.REA= 001000  | LN.OOP= 000004 | SF.MFL= 000040    |
| CF.SC= 000010   | C.FNC 000010   | FS.RET= 000000  | LN.OPE= 000001 | SF.PAC= 000020    |
| CF.SYH= 000040  | C.LIN 000006   | FS.REZ= 003000  | LN.REF= 000002 | SF.REA= 000010    |
| CF.TRN= 000100  | C.LNK 000000   | FS.RLB= 002000  | LN.SER= 000002 | SF.SER= 000001    |
| CM.CIR= 000002  | C.MOD 000011   | FS.RNG= 011000  | LN.STA= 000017 | SF.SVC= 000002    |
| CM.FMT= 100000  | C.NSP 000004   | FS.RST= 000000  | LN.SUB= 000360 | SF.UNL= 000040    |
| CM.HRD= 000002  | C.PRO 000042   | FS.RTN= 001000  | LN.TRI= 000006 | SLTMA= ***** GX   |
| CM.LIN= 000000  | C.RSV 000002   | FS.SET= 005000  | L\$ASG= 000000 | SRSTD= ***** GX   |
| CM.LOO= 000001  | C.STA 000007   | FS.SFC= 005000  | L\$DRV= 000000 | S\$WRG= 000000    |
| CM.XLQ= 000004  | C.STS 000012   | FS.SFR= 006000  | L\$P11= 000001 | S\$YSZ= 007600    |
| CP.DCF= 000040  | C.URM 177776   | FS.SFS= 004000  | L\$11R= 000000 | S.COST 000001     |
| CP.HDL= 000007  | C.XACP 000004  | FS.SPW= 040000  | L.COST 000015  | S.FLG 000000      |
| CP.PS= 177400   | C.XID 000035   | FS.STP= 000000  | L.CTL 000012   | S.LEN 000004      |
| CP.PSI= 000200  | C.XLEN 000044  | FS.STP= 002000  | L.CVA 177776   | S.NMST 000002     |
| CP.XCF= 000100  | C.XPLI 000040  | FS.STR= 001000  | L.DDM 000002   | S.OWNR 000003     |
| CP.2FR= 000030  | C.XPT 000034   | FS.TRM= 003000  | L.DDS 000004   | T\$KMG= 000000    |
| CS.ABO= 000100  | C.XSVC 000042  | FS.WLB= 001000  | L.DLC 000003   | T\$MIN= 000000    |
| CS.BRO= 000002  | C.XTC 000037   | FS.XKL= 002000  | L.DLM 000006   | V\$CTR= 001000    |
| CS.BUF= 000200  | C.X25 000036   | FS.XOF= 010000  | L.DLS 000010   | X\$DBT= 000000    |
| CS.CES= 000002  | DUMMY 000026R  | FS.XON= 007000  | L.FLG 000000   | \$AUXTB 000000RG  |
| CS.CHN= 000010  | D\$BUG= 177514 | FS.ZER= 002000  | L.KRBA 000016  | \$BF?TN= ***** GX |
| CS.CMP= 000200  | D\$ISK= 000000 | F\$LVL= 000001  | L.LEN= 000022  | \$MDCIN 000026RG  |
| CS.DCR= 000400  | D\$L11= 000001 | G\$TPP= 000000  | L.MPF 000022   | .\$\$\$\$= 000034 |
| CS.DEF= 000004  |                |                 |                |                   |

```

193 .SBTTL POWERFAIL RECOVERY DISPATCH TO DDM MODULES
194
195
196
197
198
199
200
201
202 000172 012777 177777 000000G PWRFL: MOV #1,@PWRFL ; STOP TIMERS RUNNING TO DDM PROCESSES
203 00020J 017746 000000G MOV @SLTNM,-(SP) ; SET UP COUNT OF LINES TO SCAN
204
205 000204 011603 10$: MOV (SP),R_ ; GET NEXT SYSTEM LINE #
206 000206 005303 DEC R3 ;
207 000210 006303 ASL R3 ; FORM WORD OFFSET
208 000212 067703 000000G ADD @SLTMA,R3 ; POINT INTO SYSTEM LINE INDEX TABLE
209 000216 011303 MOV (R3),R3 ; AND GET ADDRESS OF SYSTEM LINE TABLE
210 000220 005713 TST (R3) ; IS THE LINE ACTIVE?
211 000222 100010 BPL 20$; IF PL, NO
212
213 .IF DF R$$MPL
214 .IF NDF R$$PRO
215
216 BIT #F2.MP,@FMASK2 ; IS THIS A MULTIPROCESSOR?
217 BEQ 15$; BR IF NO
218 MOV L.KRBA(R3),R5 ; GET ADDRESS OF KRB
219 MOV @CPURM,-(SP) ; GET ADDRESS OF CPU URM TABLE
220 BIT K.URM(R5),@(SP)+ ; ARE WE RUNNING ON THE CORRECT PROCESSOR?
221 BEQ 20$; IF EQ, NO
222
223 15$: .ENDC
224 .ENDC
225
226 000224 016305 000004 MOV L.DDS(R3),R5 ; GET LINE TABLE ADDRESS
227 000230 116302 000002 MOV @DDM(R3),R2 ; AND DDM PDV INDEX
228 000234 016703 000000G MOV @DFNC,R3 ; POINT TO POWERFAIL FUNCTION CODE
229 000240 CALL @PLGPL ; DISPATCH TO DDM
230
231 000244 005316 20$: DEC (SP) ; REDUCE COUNT OF LINES TO SCAN
232 000246 001356 BNE 10$; LOOP IF MORE TO GO
233 000250 005726 TST (SP)+ ; CLEAN UP THE STACK
234
235 .IF NDF I$$AS
236
237 000252 012703 000302' MOV #KMCL,R3 ; POINT TO KMC LOADER TASK NAME
238 000256 CALL @SRSTD ; SCAN TO FIND TASK'S TCB
239 000262 103403 BCS 30$; IF CS, TASK NOT INSTALLED
240 000264 005001 CLR R1 ; CLEAR DEFAULT UIC
241 000266 CALL @TSKRT ; REQUEST TASK TO RUN
242
243 .ENDC
244
245 000272 017777 000000G 000000G 30$: MOV @SLTNM,@PWRFL ; START DISCONNECT NOTIFICATION ON ALL LINES
246 000300 RETURN
247
248 .IF NDF I$$AS
249 000302 043313 047574 KMCL: .RAD50 /KMCL../ ; TASK NAME OF KMC MICROCODE LOADER

```

75

.MCALL SLTDF\$,CCBDF\$,SAVRG,RESRG

76

77 000000

SLTDF\$

; DEFINE SYSTEM LINE TABLE OFFSETS

78 000000

CCBDF\$

; DEFINE CCB OFFSETS

\*\*FILE\*\*ID\*\*AXDSPP

```

AAAAAA XX XX DDDLDDDD SSSSSSSS PPPPPPPP PPPPPPPP
AAAAAA XX XX DDDDDDDD SSSSSSSS PPPPPPPP PPPPPPPP
AA AA XX XX DD DD SS PP PP PP PP
AA AA XX XX DD DD SS PP PP PP PP
AA AA XX XX DD DD SS PP PP PP PP
AA AA XX XX DD DD SS PP PP PP PP
AA AA XX XX DD DD SS PP PP PP PP
AAAAAA XX XX DD DD SS PP PP PP PP
AAAAAA XX XX DD DD SS PP PP PP PP
AA AA XX XX DD DD SS PP PP PP PP
AA AA XX XX DD DD SS PP PP PP PP
AA AA XX XX DDDDDDDD SSSSSSSS PP PP PP PP
AA AA XX XX DDDDDDDD SSSSSSSS PP PP PP PP

```

....  
....  
....  
....

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLL SSSSSSSS TT
LLLLLLLL SSSSSSSS TT

```

CESUB1 MACRO V05.03b Friday 28-Jun-85 18:20  
 Table of contents

|     |     |                                                        |
|-----|-----|--------------------------------------------------------|
| 6-  | 68  | \$CMQIN - QUEUE A CHAIN OF CCBS TO A LIST              |
| 7-  | 101 | \$CMQRM - REMOVE A CHAIN OF CCBS FROM A LIST           |
| 8-  | 142 | \$CNV18 - CONVERT TO 18-BIT UNIBUS ADDRESS             |
| 9-  | 205 | \$MVTBF - MOVE FROM MAPPED BUFFER TO UNMAPPED BUFFER   |
| 10- | 256 | \$MVFBF - MOVE FROM UNMAPPED BUFFER TO A MAPPED BUFFER |
| 11- | 310 | \$CALLX - MAPPED SUBROUTINE CALL                       |
| 12- | 365 | \$CEACC - ACCESS BLOCK IN EXTENDED POOL                |
| 12- | 366 | \$CECAC - ACCESS BLOCK IN ALTERNATE EXTENDED POOL      |
| 13- | 415 | \$PDVID - PROCESS NAME TO PDV INDEX                    |
| 14- | 450 | \$CEMUL - UNSIGNED MULTIPLY                            |
| 15- | 481 | \$CEDIV - UNSIGNED DIVISION                            |

481  
 482  
 483  
 484  
 485  
 486  
 487  
 488  
 489  
 490  
 491  
 492  
 493  
 494  
 495  
 496  
 497  
 498  
 499  
 500  
 501  
 502  
 503  
 504  
 505  
 506  
 507  
 508  
 509  
 510  
 511  
 512  
 513

000542  
 000542 010146  
 000544 010001  
 000544 005000  
 000550 071026  
 000552

```
.SBTTL $CEDIV - UNSIGNED DIVISION
*
**-- $CEDIV - UNSIGNED DIVISION
THIS SUBROUTINE PERFORMS UNSIGNED DIVISION WHERE THE RESULTS ARE LESS THAN
16 BITS. IT IS INCLUDED HERE SO THAT WE CAN CONDITIONALISE THE CODE FOR
EXTENDED INSTRUCTION SET.
NOTE THAT IT IS THE CALLERS RESPONSIBILITY TO ENSURE THAT THE RESULTS WILL
FIT IN 16 BITS.
INPUTS:
 R0 = DIVIDEND
 R1 = DIVISOR
OUTPUTS:
 R0 = QUOTIENT
 R1 = REMAINDER
;-
$CEDIV::IF DF R$$EIS
 MOV R1,-(SP) ; SAVE DIVISOR
 MOV R0,R1 ; SET UP DIVIDEND
 CLR R0 ; CLEAR HIGH ORDER WORD
 DIV (SP)+,R0 ; PERFORM THE DIVISION
 RETURN
 .IFF ; DF R$$EIS
 CALLR $DIV ; CALL EXEC DIVIDE ROUTINE
 .ENDC ; DF R$$EIS
```

```

104 .SBTTL DISPATCH A PROCESS TIMEOUT
105 :+
106 ***-$DSPTM-DISPATCH A PROCESS TIMEOUT
107 :
108 DISPATCH A PROCESS TIMEOUT AT IT'S TIMEOUT ENTRY POINT WITH A
109 SUBFUNCTION CODE OF 'FS.LTM'
110 :-
111 INPUTS:
112 R2 - PDV INDEX OF PROCESS
113 R5 - POINTER TO PROCESS LINE TABLE
114 :
115 OUTPUTS:
116 ONLY R4 WILL BE PRESERVED
117 :
118 000052 010446 $DSPTM::MOV R4,-(SP) ::: SAVE CALLER'S R4
119 000054 012703 000000G MOV #SLTMFC,R3 ::: GET ADDRESS OF FUNCTION CODE
120 000060 CALL $PDSPL ::: DISPATCH TO PROCESS
121 000064 012604 MOV (SP)+,R4 ::: RESTORE CALLER'S R4
122 000066 MIPS #0 ::: ENABLE INTERRUPTS
123 000072 RETURN
124
125 :
126 000001 .END

```

```

294
295 000324 $SYNB::
296 000324 000000 .WORD 0
297 000326 000352' .WORD $SYNBF
298 000330 000010 .WORD $SYNCT
299
300 000332 $PADB::
301 000332 000000 .WORD 0
302 000334 000340' .WORD $PADBF
303 000336 000012 .WORD $PADKL
304
305 000340 $PADBF::
306 000012 .REPT $PADKL
307 .BYTE $PAD
308 .ENDR
309 000352 $SYNBF::
310 000010 .REPT $SYNCT
311 .BYTE $SYNC
312 .ENDR
313
314 ;
315 ; COMM EXEC FREE SPACE POINTER
316 ;
317 .IF NDF R$$MPL
318 000000 .PSECT $$$POL
319
320 000000 000003 .WORD 3 ; ROUNDING FACTOR
321 000002 000000 $CEAVL:: .WORD 0 ; POINTER TO FIRST FREE BLOCK
322 000004 000000 .WORD 0
323
324 000000 .PSECT
325
326 .ENDC ; NDF R$$MPL
327
328 .ENDC ; DF M$$NET
329
330 .IF DF R$$MPL
331 $NTEND:: ; END OF ARE TO DEALLOCATE IN SET /NOCEX
332 ; MCR COMMAND
333 .ENDC ; R$$MPL
334
335
336 000001 .END
337

```



```

109 .SBTTL $CLCRC - CALCULATE CRC-16 ON A TRANSMIT CHAIN
110
111 ;+
112 ;**-$CLCRC-CALCULATE CRC (CRC16) ON A TRANSMIT CHAIN
113
114 INPUTS:
115 R4 - ADDRESS OF FIRST CCB IN CHAIN
116 C.LNK - ADDRESS OF NEXT CCB IN CHAIN (0 MARKS END)
117 .BUF - DOUBLE WORD ADDRESS OF BUFFER
118 C.CNT - LENGTH OF BUFFER
119 C.FLG - BUFFER FLAGS
120
121 OUTPUTS:
122 CRC IS STORED AT THE END OF EACH BUFFER AND THE BYTE
123 COUNT IS UPDATED.
124
125 REGISTERS MODIFIED:
126 R2, R3
127
128 000006 $CLCRC::
129 .IF DF M$$MGE
130 .IF NDF I$$AS
131
132 000006 017746 177770 MOV @K$AR5,-(SP) ; SAVE CALLER'S MAPPING
133
134 .IFF ; I$$AS
135
136 MOV #KP.AR3,-(SP) ; ASSUME KERNEL MODE
137 BIT #140000,PS.EXP ; IS IT ?
138 BEQ 5$; YES - BR
139 MOV #UPAR0+6,(SP) ; NO - MUST BE USER MODE
140 5$: MOV @ (SP),-(SP) ; SAVE CURRENT MAPPING
141
142 .ENDC ; I$$AS
143 .ENDC ; M$$MGE
144
145 000012 SAVRG <R0,R1,R4> ; SAVE REGISTERS
146 000020 005001 CLR R1 ; INITIALISE CRC ACCUMULATOR
147
148 000022 010403 10$: MOV R4,R3 ; COPY CCB ADDRESS
149 000024 062703 000014 ADD #C.BUF,R3 ; AND POINT TO BUFFER DESCRIPTOR
150
151 .IF DF M$$MGE
152 IF NDF I$$AS
153
154 000030 012377 177746 MOV (R3)+,@K$AR5 ; MAP TO THE DATA
155
156 .IFF ; I$$AS
157
158 MOV (R3)+,@10(SP) ; MAP TO THE DATA
159
160 .ENDC ; I$$AS
161 .IFF ; M$$MGE
162
163 TST (R3)+ ; SKIP OVER RELOCATION BIAS
164
165 .IFF ; M$$MGE

```

STCR2 MACRO V05.03b Friday 28-Jun-85 18:21 Page 11-6

## Symbol table

|           |          |           |         |           |          |           |          |           |           |
|-----------|----------|-----------|---------|-----------|----------|-----------|----------|-----------|-----------|
| A\$\$CHK= | 000000   | CS.ABO=   | 000100  | C.STA     | 000007   | FS.RLB=   | 002000   | M\$\$NET= | 000000    |
| A\$\$CPS= | 000000   | CS.BRO=   | 000302  | C.STS     | 000012   | FS.RNG=   | 011000   | M\$\$OVR= | 000000    |
| A\$\$PRI= | 000000   | CS.BUF=   | 000200  | C.URM     | 177776   | FS.RST=   | 000000   | N\$\$ACC= | 000001    |
| A\$\$TRP= | 000000   | CS.CES=   | 000002  | C.XACP    | 000004   | FS.RTN=   | 001000   | N\$\$BUF= | 000001    |
| BUFUMP=   | 172354   | CS.CHN=   | 000010  | C.XID     | 000035   | FS.SET=   | 005000   | N\$\$LDV= | 000001    |
| CB.CCB=   | 000002   | CS.CMP=   | 000200  | C.XLEN    | 000044   | FS.SFC=   | 005000   | N\$\$MCP= | 000001    |
| CB.DDM=   | 000040   | CS.DCR=   | 000400  | C.XPLI    | 000040   | FS.SFR=   | 006000   | N\$\$MLL= | 000001    |
| CB.DLC=   | 000020   | CS.DEF=   | 000004  | C.XPT     | 000034   | FS.SFS=   | 004000   | N\$\$MOV= | 000010    |
| CB.RDB=   | 000004   | CS.DEV=   | 000002  | C.XSVC    | 000042   | FS.SPW=   | 040000   | N\$\$NCT= | 000001    |
| CB.SDB=   | 000010   | CS.DIS=   | 000040  | C.XTC     | 000037   | FS.STM=   | 000000   | N\$\$PEM= | 000001    |
| CB.SLI=   | 000100   | CS.ENA=   | 000001  | C.X25     | 000036   | FS.STP=   | 002000   | PIRQ      | = 177772  |
| CB.XLB=   | 000001   | CS.ENB=   | 000020  | DDB       | = 000010 | FS.STR=   | 001000   | PMODE     | = 030000  |
| CC.LLC=   | 000200   | CS.ERR=   | 100000  | D\$\$BUG= | 177514   | FS.TRM=   | 003000   | PR0       | = 000000  |
| CE.ABO=   | 100362   | CS.FTL=   | 001000  | D\$\$ISK= | 000000   | FS.WLB=   | 001000   | PR1       | = 000040  |
| CE.DAO=   | 100346   | CS.HCR=   | 000001  | D\$\$L11= | 000001   | FS.XKL=   | 002000   | PR2       | = 000100  |
| CE.DIS=   | 100366   | CS.HFE=   | 002000  | D\$\$YNC= | 000000   | FS.XOF=   | 010000   | PR3       | = 000140  |
| CE.ERR=   | 100370   | CS.LST=   | 040000  | D\$\$YNM= | 000000   | FS.XON=   | 007000   | PR4       | = 000200  |
| CE.ILN=   | 100350   | CS.MTL=   | 004000  | E\$\$XPR= | 000000   | FS.ZER=   | 002000   | PR5       | = 000240  |
| CE.LTO=   | 100356   | CS.RNG=   | 000010  | FC.CCP=   | 000020   | F\$\$LVL= | 000001   | PR6       | = 000300  |
| CE.MOP=   | 100372   | CS.ROV=   | 000004  | FC.CTL=   | 000006   | G\$\$TPP= | 000000   | PR7       | = 000340  |
| CE.NTE=   | 100361   | CS.RSN=   | 010000  | FC.KCP=   | 000016   | G\$\$TSS= | 000000   | PS        | = 177776  |
| CE.RTE=   | 100376   | CS.SHU=   | 000001  | FC.KIL=   | 000004   | G\$\$TTK= | 000000   | P\$\$P45= | 000000    |
| CE.SRC=   | 100364   | CS.SID=   | 000002  | FC.MAN=   | 000024   | G\$\$WRD= | 000000   | P\$\$WRD= | 000000    |
| CE.GTP=   | 100352   | CS.STR=   | 000004  | FC.MLD=   | 000026   | I\$\$RAR= | 000000   | Q\$\$OPT= | 000010    |
| CE.IME=   | 100354   | CS.SUC=   | 000001  | FC.PCT=   | 000030   | I\$\$RDN= | 000000   | R\$\$DER= | 000000    |
| CE.TMO=   | 100374   | CS.TMO=   | 020000  | FC.PWR=   | 000022   | KGCSR     | = 170700 | R\$\$EIS= | 000000    |
| CE.UNS=   | 100344   | CS.XUR=   | 000004  | FC.RCE=   | 000002   | KGINIT    | = 000111 | R\$\$K11= | 000001    |
| CF.CHN=   | 000001   | CTABL     | 000256R | FC.RCP=   | 000014   | KGLDBC    | = 000133 | R\$\$SNO= | 000000    |
| CF.EQM=   | 000004   | C\$\$CKP= | 000000  | FC.TIM=   | 000010   | KISAR0=   | 172340   | R\$\$11M= | 000000    |
| CF.HDR=   | 000020   | C\$\$ORE= | 000400  | FC.XCP=   | 000012   | KISAR5=   | ***** GX | SEN       | = 000100  |
| CF.LB     | = 100000 | C\$\$RSH= | 177564  | FC.XME=   | 000000   | KISAR6=   | 172354   | STCR2     | = 000162R |
| CF.LIN=   | 000002   | C.ADD     | 000034  | FS.AST=   | 000000   | KSAR5     | 000002R  | SWR       | = 177570  |
| CF.SOM=   | 000010   | C.BID     | 000003  | FS.CIB=   | 002000   | KSAR6     | 000004R  | S\$\$WRG= | 000000    |
| CF.SYN=   | 000040   | C.BUF     | 000014  | FS.CRA=   | 001000   | K\$\$CNT= | 177546   | S\$\$YSZ= | 007600    |
| CF.TRN=   | 000100   | C.BUF1    | 000014  | FS.DIS=   | 013000   | K\$\$CSR= | 177546   | TPS       | = 177564  |
| CLRG      | = 000020 | C.BUF2    | 000024  | FS.DVC=   | 001000   | K\$\$LDC= | 000000   | T\$\$KMG= | 000000    |
| CMODE     | = 140000 | C.CNT     | 000020  | FS.ENB=   | 012000   | K\$\$TPS= | 000074   | T\$\$MIN= | 000000    |
| CM.CIR=   | 000002   | C.CNT1    | 000020  | FS.EXI=   | 001000   | L^L^LP    | = 000000 | UBMPR     | = 170200  |
| CM.FMT=   | 100000   | C.CNT2    | 000030  | FS.GET=   | 006000   | LRC16     | = 000003 | UISAR0=   | 177640    |
| CM.HRD=   | 000002   | C.FLG     | 000022  | FS.HLT=   | 000000   | L\$\$ASG= | 000000   | UISAR1=   | 177642    |
| CM.LIN=   | 000000   | C.FLG1    | 000022  | FS.INI=   | 000000   | L\$\$DRV= | 000000   | V\$\$CTR= | 001000    |
| CM.LOO=   | 000001   | C.FLG2    | 000032  | FS.KIL=   | 000000   | L\$\$P11= | 000001   | X\$\$DBT= | 000000    |
| CM.XLO=   | 000004   | C.FNC     | 000010  | FS.LCL=   | 100000   | L\$\$11R= | 000000   | \$CLCRC   | 000006RG  |
| CP.DCF=   | 000040   | C.LIN     | 000006  | FS.LTM=   | 001000   | MPAR      | = 172100 | \$STCRP   | 000122RG  |
| CP.HDL=   | 000007   | C.LNK     | 000000  | FS.MNT=   | 004000   | MPCSR     | = 177746 | \$STCR1   | 000124RG  |
| CP.PS     | = 177400 | C.MOD     | 000011  | FS.MSN=   | 014000   | M\$\$CRB= | 000124   | \$STCVL=  | 000002 G  |
| CP.PSI=   | 000200   | C.NSP     | 000004  | FS.REA=   | 001000   | M\$\$CRX= | 000000   | \$STCVT   | 000000RG  |
| CP.XCF=   | 000100   | C.PRO     | 000042  | FS.RET=   | 000000   | M\$\$FCS= | 000000   | .BASEB    | = 140000  |
| CP.2FR=   | 000030   | C.RSV     | 000002  | FS.REZ=   | 003000   | M\$\$MGE= | 000000   | .\$\$\$.  | = 000034  |
| CRC16     | = 000001 |           |         |           |          |           |          |           |           |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
 000356 001 (RW,I,LCL,REL,CON)  
 Errors detected: 0

\*\*\* Assembler statistics

```

166
167 000034 012302 MOV (R3)+,R2 ; GET VIRTUAL ADDRESS OF BUFFER
168
169 .IFT ; M$$MGE
170 .IF NDF 1$$AS
171
172 000036 022702 140000 CMP #140000,R2 ; IS BUFFER MAPPED BY AND APR < 6?
173 000042 101002 BHI 20$; IF HI, YES
174 000044 162732 020000 SUB #20000,R2 ; MAP BUFFER THROUGH APR5
175
176 .IFF ; 1$$AS
177 000050 012300 CMP #60000,R2 ; IS BUFFER MAPPED BY AN APR 3 ?
178 BLOS 20$; IF LO OR SAME, NO - MUST BE APR 3
179 ADD #20000,R2 ; MAP BUFFER THROUGH APR3
180 .ENDC ; 1$$AS
181 000050 20$: .ENDC ; M$$MGE
182
183
184 000050 012300 MOV (R3)+,R0 ; GET LENGTH OF SEGMENT
185 000052 CALL STCR2 ; COMPUTE CRC ON SEGMENT
186 000056 032713 000004 BIT #CF.EOM,(R3) ; LAST SEGMENT IN MESSAGE?
187 000062 001407 BEQ 30$; IF EQ, NO
188
189 000064 060002 ADD R0,R2 ; POINT TO END OF MESSAGE
190 000066 110122 MOVW R1,(R2)+ ; STORE CRC FOLLOWING DATA
191 000070 000301 SWAB R1 ; ...
192 000072 110122 MOVW R1,(R2)+ ; ...
193 000074 062743 000002 ADD #2,-(R3) ; UPDATE COUNT IN CCB
194 000100 005001 CLR R1 ; RE-INITIALISE CRC ACCUMULATOR
195
196 000102 011404 30$: MOV (R4),R4 ; GET NEXT CCB IN CHAIN
197 000104 001346 BNE 10$; AND COMPUTE IT'S CRC
198
199 000106 RESRG <R4,R1,R0> ; RESTORE REGISTERS
200
201 .IF DF M$$MGE
202 .IF NDF 1$$AS
203
204 000114 012677 177662 MOV (SP)+,@KSAR5 ; RESTORE MAPPING
205
206 .IFF ; 1$$AS
207
208 MOV (SP)+,@(SP)+ ; RESTORE MAPPING
209
210 .ENDC ; 1$$AS
211 .ENDC ; M$$MGE
212
213 000120 RETURN

```

\*\*wjd01\*\*

STCRCF MACRO V05.03b Friday 28-Jun-85 18:22 Page 12-7  
Symbol table

Work file reads: 0  
Work file writes: 0  
Size of work file: 15190 words ( 60 Pages)  
Size of core pool: 16552 words ( 63 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:16.30  
SY:STCRCF.V2,[130,134]STCRCF/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]V2,STCRC

STCRCK MACRO V05.03b Friday 28-Jun-85 18:22 Page 10  
 \$STCRC - CALCULATE CRC ON BLOCK OF DATA

```

215 .SBTTL $STCRC - CALCULATE CRC ON BLOCK OF DATA
216
217 **-$STCRC-CALCULATE CRC ON BLOCK OF DATA
218 **-$STCR1-ALTERNATE ENTRY (R1 - PARTIAL CRC)
219
220 CALCULATE THE CRC ON A BLOCK OF DATA.
221
222 INPUTS:
223
224 R0 - # OF BYTES IN DATA BLOCK
225 R1 - PARTIAL CRC OR 0
226 R2 - POINTER TO DATA BLOCK
227 R4 - CCB DESCRIBING DATA BUFFER
228
229 OUTPUTS:
230 R1 - UPDATED CRC VALUE
231
232 000122 005001 $STCRC::CLR R1 ; INITIALISE CRC ACCUMULATOR
233
234 000124 $STCR1::
235 .IF DF M$$MGE
236 .IF NDF I$$AS
237
238 000124 017746 177652 MOV @K$AR5,-(SP) ; SAVE CURRENT MAPPING
239 000130 016477 000014 177644 MOV C.BUF(R4),@K$AR5 ; AND MAP TO BUFFER
240 000136 022702 140000 CMP #140000,R2 ; IS BUFFER MAPPED BY APR > 6?
241 000142 101002 BHI 10$; IF HI, NO
242 000144 162702 020000 SUB #20000,R2 ; MAP BUFFER USING APR5
243
244 .IFF ; I$$AS
245
246 MOV #KP.AR3,-(SP) ; ASSUME KERNEL MODE
247 BIT #140000,PS.EXP ; KERNEL MODE ?
248 BEQ 5$; YES - BR
249 MOV #UPAR0+6,(SP) ; NO - MUST BE USER MODE
250 5$: MOV @K$AR5,-(SP) ; SAVE CURRENT MAPPING
251 MOV C.BUF(R4),@2(SP) ; MAP TO DATA BUFFER
252 CMP #60000,R2 ; IS BUFFER MAPPED BY APR 2?
253 BLOS 10$; IF LO OR SAME, NO
254 ADD #20000,R2 ; MAP BUFFER USING APR3
255
256 .ENDC ; I$$AS
257 000150 10$: .ENDC ; M$$MGE
258
259 000150 CALL STCR2 ; CALCULATE CRC
260
261 .IF DF M$$MGE
262 .IF NDF I$$AS
263
264 000154 012677 177622 MOV (SP)+,@K$AR5 ; RESTORE MAPPING
265
266 .IFF
267
268 MOV (SP)+,@(SP)+ ; RESTORE MAPPING
269
270 .ENDC ; I$$AS
271

```

STCRCK CREATED BY MACRO ON 28-JUN-85 AT 18:22

PAGE 1 C 10

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE       | REFERENCES                                    |
|---------|-------------|-----------------------------------------------|
| BUFUMP  | = 172354    | #6-68                                         |
| CF.EOM  | = 000004    | 9-186                                         |
| CLRKG   | = 000020    | 7-96 7-98                                     |
| CMODE   | = 140000    | #6-68                                         |
| CRC16   | = 000001    | #7-93 7-99                                    |
| C.BUF   | = 000014    | 9-149 10-239                                  |
| DDB     | = 000010    | #7-95 7-98                                    |
| I\$SAS  | = *****     | 6-68 7-99 11-312 11-316 10-236 10-263         |
| KGCSR   | = 170700    | #7-92 9-130 9-152 9-170 9-202                 |
| KGINIT  | = 000111    | #7-99 11-295 11-296 11-305                    |
| KGLDBC  | = 000133    | #7-98 11-299 11-334                           |
| KISARO  | = 172340    | #6-68                                         |
| KISAR5  | = ***** GX  | 8-104                                         |
| KISAR6  | = 172354    | #6-68 8-105                                   |
| KSAR5   | = 000002 R  | #8-104 9-132 9-154 9-204 10-238 10-239 10-265 |
| KSAR6   | = 000004 R  | #8-105                                        |
| K\$G11  | = 000001    | 5-1 11-293 12-431                             |
| LRC16   | = 000003    | #7-94 7-98                                    |
| L\$S11  | = *****     | 6-73                                          |
| MPAR    | = 172100    | #6-68                                         |
| MPCSR   | = 177746    | #6-68                                         |
| M\$SMGE | = 000000    | 6-68 6-73 9-129 9-151 9-201 10-235 10-262     |
| PIRO    | = 177772    | #6-68                                         |
| PMODE   | = 030000    | #6-68                                         |
| PRO     | = 000000    | #6-68                                         |
| PR1     | = 000040    | #6-68                                         |
| PR2     | = 000100    | #6-68                                         |
| PR3     | = 000140    | #6-68                                         |
| PR4     | = 000200    | #6-68                                         |
| PR5     | = 000240    | #6-68                                         |
| PR6     | = 000300    | #6-68                                         |
| PR7     | = 000340    | #6-68                                         |
| PS      | = 177776    | #6-68                                         |
| P\$S34  | = *****     | 6-73                                          |
| P\$S40  | = *****     | 6-73                                          |
| P\$S45  | = *****     | 6-73                                          |
| P\$S70  | = *****     | 6-73                                          |
| R\$SEIS | = 000000    | #6-75 6-79 6-68 6-68                          |
| R\$EMPL | = *****     | 6-68                                          |
| R\$S11D | = *****     | 6-68                                          |
| SEN     | = 000100    | #7-97 7-98 7-99                               |
| STCR2   | = 000162 R  | 9-185 10-260 #11-291                          |
| SWR     | = 177570    | #6-68                                         |
| TPS     | = 177564    | #6-68                                         |
| UBMPR   | = 170200    | #6-68                                         |
| UISARO  | = 177640    | #6-68                                         |
| UISAR1  | = 177642    | #6-68                                         |
| \$CLCRC | = 000006 RG | #9-128                                        |
| \$STCRC | = 000122 RG | #10-232                                       |
| \$STCR1 | = 000124 RG | #10-234                                       |
| \$STCVL | = 000002 G  | #8-107                                        |
| \$STCVT | = 000000 RG | #8-103 8-107                                  |

AXBFR CREATED BY MACRO ON 28-JUN-85 AT 18:28

PAGE 1 C 11

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL | VALUE    | REFERENCES             |
|--------|----------|------------------------|
| BUFUMP | = 172354 | #5-61                  |
| CB.RDB | = 000004 | 6-94                   |
| CF.LB  | = 100000 | 6-91                   |
| C.MODE | = 140000 | #5-61                  |
| C.BID  | 000003   | *6-94                  |
| C.CNT  | 000020   | *6-92                  |
| C.CNT2 | 000030   | *6-93                  |
| C.FLG  | 000022   | *6-91                  |
| C.FNC  | 000010   | 6-155 *6-172           |
| C.LIN  | 000006   | *6-151                 |
| F.HCE  | = 000002 | 6-172                  |
| FS.RTN | = 001000 | 6-172                  |
| ISSAS  | = *****  | 5-61 6-100 6-160 6-180 |
| KISAR0 | = 172340 | #5-61                  |
| KISAR6 | = 172354 | #5-61                  |
| LF.ACT | = 100000 | #5-60                  |
| LF.BRO | = 000400 | #5-60                  |
| LF.BWT | = 000007 | #5-60 6-129            |
| LF.ENA | = 002000 | #5-60                  |
| LF.LPB | = 001000 | #5-60                  |
| LF.MDC | = 000100 | #5-60                  |
| LF.MFL | = 004000 | #5-60                  |
| LF.MTP | = 000020 | #5-60                  |
| LF.PAC | = 000200 | #5-60                  |
| LF.RDY | = 040000 | #5-60                  |
| LF.REA | = 010000 | #5-60                  |
| LF.SER | = 000040 | #5-60                  |
| LF.TIM | = 000010 | #5-60                  |
| LF.UNL | = 020000 | #5-60                  |
| LF.X2P | = 000000 | #5-60                  |
| LN.CLO | = 000000 | #5-60                  |
| LN.DUM | = 000005 | #5-60                  |
| LN.LOA | = 000004 | #5-60                  |
| LN.LOO | = 000003 | #5-60                  |
| LN.OAJ | = 000003 | #5-60                  |
| LN.OFF | = 000001 | #5-60                  |
| LN.ON  | = 000000 | #5-60                  |
| LN.OOP | = 000004 | #5-60                  |
| LN.OPE | = 000001 | #5-60                  |
| LN.REF | = 000002 | #5-60                  |
| LN.SER | = 000002 | #5-60                  |
| LN.STA | = 000017 | #5-60                  |
| LN.SUB | = 000360 | #5-60                  |
| LN.TRI | = 000006 | #5-60                  |
| L.COST | 000015   | #5-60                  |
| L.CTL  | 000012   | #5-60                  |
| L.CVA  | 177776   | #5-60                  |
| L.DDM  | 000002   | 6-153                  |
| L.DDS  | 000004   | 6-152                  |
| L.DLC  | 000003   | #5-60                  |
| L.DLM  | 000006   | #5-60                  |
| L.DLS  | 000010   | #5-60                  |

\*\*FILE\*\*ID\*\*AXDSP

```

AAAAAA XX XX DDDDDDD SSSSSSS PPPPPPP
AAAAAA XX XX DDDDDDD SSSSSSS PPPPPPP
AA AA XX DD SS PP PP
AA AA XX DD SS PP PP
AA AA XX DD SS PP PP
AA AA XX DD SS PP PP
AA AA XX DD SS PP PP
AA AA XX DD SS PP PP
AAAAAAA XX XX DD SS PP
AAAAAAA XX XX DD SS PP
AA AA XX DD SS PP
AA AA XX DD SS PP
AA AA XX DD SS PP
AA AA XX DD SS PP
AA AA XX DD SS PP

```

```

LL SSSSSSS TTTTTTTTT
LL SSSSSSS TTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLL SSSSSSS TT
LLLLLLLLL SSSSSSS TT

```



AXDSP - AUXILLIARY PROCESS DISP MACRO V05.03b Tuesday 03-Sep-85 10:58 Page 9-3  
Symbol table

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000244 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 16127 Words ( 63 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:17.28  
DB2:AXDSP.T47,[131,134]AXDSP/CR/-SP=DB2:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]AXDSP

AXDSPB - AUXILLIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 10:58 Page 9-1  
POWERFAIL RECOVERY DISPATCH TO DDM MODULES

250  
251  
252

.ENDC

.ENDC

```

80
81 ; AUXILARY PROCESS DISPATCH TABLE
82 ;
83
84 $AUXTB::
85 .WORD .+1 ; TRANSMIT ENABLE (NOP)
86 .WORD $BFRTN ; RECEIVE ENABLE
87 .WORD .+1 ; KILLIO (NOP)
88 .IF DF X$$MDC ;
89 .WORD MDMCTL ; MODEM CONTROL CONTROL ROUTINE
90 .IF DF P$$RFL ;
91 .WORD PWFAIL ; PERFORM POWERFAIL RECOVERY
92 .IF
93 .WORD MDMSCN ; TIMEOUT ENTERS SCAN ROUTINE
94 .ENDC
95 .IF
96 .WORD .+1 ; CONTROL ENABLE (NOP)
97 .IF DF P$$RFL ;
98 .WORD PWFAIL ; PERFORM POWERFAIL RECOVERY
99 .IF
100 .WORD DUMMY ; TIMEOUT (NOP)
101 .ENDC
102 .ENDC
103 .WORD .+1 ; TRANSMIT COMPLETE (NOP)
104 .WORD .+1 ; RECEIVE COMPLETE (NOP)
105 .WORD .+1 ; KILL COMPLETE (NOP)
106 .WORD CCBRET ; CONTROL COMPLETE - RELEASE CCB
107 .IF DF P$$RFL ;
108 .WORD PWRFL ; POWERFAIL DISPATCH TO DDM MODULES
109 .IF
110 .WORD DUMMY ; POWERFAIL DISPATCH TO DDM MODULES (NOP)
111 .ENDC
112 .WORD NMCMP ; NETWORK MANAGEMENT COUNTER COMPLETION
113
114 .IF NDF X$$MDC & R$$11D & I$$AS
115 $MDCIN::
116 .ENDC
117
118 DUMMY: RETURN
119 CCBRET: CALLR @CCBRT
120 000026
120 000030

```

AXDSPP - AUXILLIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 C 16  
Table of contents

|     |     |                                            |
|-----|-----|--------------------------------------------|
| 8-  | 123 | POWERFAIL RECOVERY ROUTINE                 |
| 9-  | 193 | POWERFAIL RECOVERY DISPATCH TO DDM MODULES |
| 10- | 254 | NETWORK MANAGEMENT COUNTER COMPLETION      |

1 .IIF NDF RSSEIS .TITLE CESUB  
2 .IIF DF RSSEIS .TITLE CESUB1  
3 .IDENT /V05.00/  
4

5 : COPYRIGHT (C) 1978,1979,1980,1981, 1982, 1983, 1985 BY  
6 : DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.  
7

8 : THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
9 : ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
10 : INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
11 : COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
12 : OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
13 : TRANSFERRED.  
14

15 : THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
16 : AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
17 : CORPORATION.  
18

19 : DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
20 : SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
21

22 : MODULE DESCRIPTION  
23

24 : CEX SUBROUTINES  
25

26 : DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING  
27

28 : IDENT HISTORY:  
29

- 30 : 1.00 10-FEB-78  
31 : VERSION 2.0 RELEASE  
32 :  
33 : 2.00 14-DEC-79  
34 : DECNET-11M/S V3.0  
35 : DECNET-11M-PLUS V1.0  
36 :  
37 : 3.00 16-APR-82  
38 : DECNET-11M V3.1  
39 : DECNET-11M-PLUS V1.1  
40 :  
41 : 4.00 07-NOV-83  
42 : DECNET-11M V4.0  
43 : DECNET-11M-PLUS V2.0  
44 :  
45 : 5.00 22-JUL-85  
46 : DECnet-11M/S V4.2  
47 : DECnet-11M-Plus V3.0  
48 : DECnet-Micro/R SX V1.0  
49 :  
50 :  
51 :  
52 :  
53 :  
54 :  
55 :  
56 :  
57 :

```
515 .IF DF M$$PRO
516
517 .SBTTL $MPLCK - MULTI-PROCESSOR LOCK ROUTINE
518
519 +
520 **-$MPLCK-MULTI-PROCESSOR LOCK ROUTINE
521 :
522 : THIS ROUTINE WILL PERFORM A SPIN LOCK ON THE COMMEXEC LOCK ($CRESL)
523 : AND THEN CO-CALL THE CALLER. ON RETURN IT WILL UNLOCK THE COMMEXEC
524 : LOCK AND RETURN. NOTE THAT WE PERFORM THE LOCK OPERATION OURSELVES
525 : SO THAT WE CAN CORRECTLY MANIPULATE THE CACHE.
526 :
527 -$MPLCK::ASRB $CRESL ; SPIN ON THE LOCK
528 BCC $MPLCK ; UNTIL WE HAVE ACCESS
529 CACHE$ FLUSH ; FLUSH THE CACHE OF STALE DATA
530
531 CALL @($P)+ ; CO-CALL THE CALLER BACK
532
533 MOVB #1,$CRESL ; RESET THE LOCK
534
535 RETURN
```

## Symbol table

|         |        |          |        |          |          |          |        |          |          |
|---------|--------|----------|--------|----------|----------|----------|--------|----------|----------|
| ASSCHK= | 000000 | CS.ENB=  | 000020 | FC.KIL=  | 000004   | K\$STPS= | 000074 | N\$SACC= | 000001   |
| ASSCPS= | 000000 | CS.ERR=  | 100000 | FC.MAN=  | 000024   | LD\$LP=  | 000000 | N\$SBUF= | 000001   |
| ASSPRI= | 000000 | CS.FTL=  | 001000 | FC.MLD=  | 000026   | LF.ACT=  | 100000 | N\$SDV=  | 000001   |
| ASSTRP= | 000000 | CS.HCR=  | 000001 | FC.PCT=  | 000030   | LF.BRO=  | 000400 | N\$SMCP= | 000001   |
| CB.CCB= | 000002 | CS.HFE=  | 002000 | FC.PWR=  | 000022   | LF.BWT=  | 000007 | N\$SMML= | 000001   |
| CB.DDM= | 000040 | CS.LST=  | 000000 | FC.RCE=  | 000002   | LF.ENA=  | 002000 | N\$SMOV= | 000010   |
| CB.DLC= | 000020 | CS.MTL=  | 004000 | FC.RCP=  | 000014   | LF.LPB=  | 001000 | N\$SNCT= | 000001   |
| CB.RDB= | 000034 | CS.RNG=  | 000010 | FC.TIM=  | 000010   | LF.MDC=  | 000100 | N\$SPEM= | 000001   |
| CB.SDB= | 000010 | CS.ROV=  | 000004 | FC.XCP=  | 000012   | LF.MFL=  | 004000 | PR7=     | ***** GX |
| CB.SLI= | 000100 | CS.RSN=  | 010000 | FC.XME=  | 000000   | LF.MTP=  | 000020 | PS=      | ***** GX |
| CB.XLB= | 000001 | CS.SHU=  | 000001 | FS.AST=  | 000000   | LF.PAC=  | 000200 | P\$P45=  | 000000   |
| CC.LLC= | 000200 | CS.SID=  | 000002 | FS.CIB=  | 002000   | LF.RDY=  | 040000 | P\$SWRD= | 000000   |
| CE.ABO= | 100362 | CS.STR=  | 000004 | FS.CRA=  | 001000   | LF.REA=  | 010000 | Q\$SOPT= | 000010   |
| CE.DAO= | 100346 | CS.SUC=  | 000001 | FS.DIS=  | 013000   | LF.SER=  | 000040 | R\$SDCR= | 000000   |
| CE.DIS= | 100366 | CS.TMO=  | 020000 | FS.DVC=  | 001000   | LF.TIM=  | 000010 | R\$SK11= | 000001   |
| CE.ERR= | 100370 | CS.XUR=  | 000004 | FS.ENB=  | 012000   | LF.UNL=  | 020000 | R\$SSND= | 000000   |
| CE.ILN= | 100350 | C\$CKP=  | 000000 | FS.EXI=  | 001000   | LF.X2P=  | 000000 | R\$S11M= | 000000   |
| CE.LTO= | 100356 | C\$QRE=  | 000400 | FS.GET=  | 006000   | LN.CLO=  | 000000 | SF.ACT=  | 000200   |
| CE.MOP= | 100372 | C\$RSH=  | 177564 | FS.HLT=  | 000000   | LN.DUM=  | 000005 | SF.ENA=  | 000100   |
| CE.NTE= | 100361 | C.ADD=   | 000034 | FS.INI=  | 000000   | LN.LOA=  | 000004 | SF.LPB=  | 000004   |
| CE.RTE= | 100376 | C.BID=   | 000003 | FS.KIL=  | 000000   | LN.LOO=  | 000003 | SF.MFL=  | 000040   |
| CE.SRC= | 100364 | C.BUF=   | 000014 | FS.LCL=  | 100000   | LN.OAU=  | 000003 | SF.PAC=  | 000020   |
| CE.STP= | 100352 | C.BUF1=  | 000014 | FS.LTM=  | 001000   | LN.OFF=  | 000001 | SF.REA=  | 000010   |
| CE.TME= | 100354 | C.BUF2=  | 000024 | FS.MNT=  | 004000   | LN.ON=   | 000000 | SF.SER=  | 000001   |
| CE.TMO= | 100374 | C.CNT=   | 000020 | FS.MSN=  | 014000   | LN.OOP=  | 000004 | SF.SVC=  | 000002   |
| CE.UNS= | 100344 | C.CNT1=  | 000020 | FS.REA=  | 001000   | LN.OPE=  | 000001 | SF.UNL=  | 000040   |
| CF.CHN= | 000001 | C.CNT2=  | 000030 | FS.RET=  | 000000   | LN.REF=  | 000002 | S\$SWRG= | 000000   |
| CF.EOM= | 000004 | C.FLG=   | 000022 | FS.REZ=  | 003000   | LN.SER=  | 000002 | S\$SYZ=  | 007600   |
| CF.HDR= | 000020 | C.FLG1=  | 000022 | FS.RLB=  | 002000   | LN.STA=  | 000017 | S.COST=  | 000001   |
| CF.LB=  | 100000 | C.FLG2=  | 000032 | FS.RNG=  | 011000   | LN.SUB=  | 000360 | S.FLG=   | 000000   |
| CF.LIN= | 000002 | C.FNC=   | 000010 | FS.RST=  | 000000   | LN.TRI=  | 000006 | S.LEN=   | 000004   |
| CF.SOM= | 000010 | C.LIN=   | 000006 | FS.RTN=  | 001000   | L\$ASG=  | 000000 | S.NMST=  | 000002   |
| CF.SYN= | 000040 | C.LNK=   | 000000 | FS.SET=  | 005000   | L\$DRV=  | 000000 | S.OWNR=  | 000003   |
| CF.TRN= | 000100 | C.MOD=   | 000011 | FS.SFC=  | 005000   | L\$P11=  | 000001 | T\$KMG=  | 000000   |
| CM.CIR= | 000002 | C.NSP=   | 000004 | FS.SFR=  | 006000   | L\$P1R=  | 000000 | T\$MIN=  | 000000   |
| CM.FMT= | 100000 | C.PRO=   | 000042 | FS.SFS=  | 004000   | L.COST=  | 000015 | V\$CTR=  | 001000   |
| CM.HRD= | 000002 | C.RSV=   | 000002 | FS.SPW=  | 040000   | L.CTL=   | 000012 | X\$SDBT= | 000000   |
| CM.LIN= | 000000 | C.STA=   | 000007 | FS.STM=  | 000000   | L.CVA=   | 177776 | ZF.COU=  | 001000   |
| CM.LOO= | 000001 | C.STS=   | 000012 | FS.STP=  | 002000   | L.DDM=   | 000002 | ZF.DDM=  | 000001   |
| CM.XLO= | 000004 | C.URM=   | 177776 | FS.STR=  | 001000   | L.DDS=   | 000004 | ZF.DIA=  | 004000   |
| CP.DCF= | 000040 | C.XACP=  | 000004 | FS.TRM=  | 003000   | L.DLC=   | 000003 | ZF.DLC=  | 000002   |
| CP.HDL= | 000007 | C.XID=   | 000035 | FS.WLB=  | 001000   | L.DLM=   | 000006 | ZF.DVP=  | 100000   |
| CP.PS=  | 177400 | C.XLEN=  | 000044 | FS.XKL=  | 002000   | L.DLS=   | 000010 | ZF.INI=  | 040000   |
| CP.PSI= | 000200 | C.XPLI=  | 000040 | FS.XOF=  | 010000   | L.FLG=   | 000000 | ZF.KMX=  | 000020   |
| CP.XCF= | 000100 | C.XPT=   | 000034 | FS.XON=  | 007000   | L.KRBA=  | 000016 | ZF.LLC=  | 000004   |
| CP.ZFR= | 000030 | C.XSVC=  | 000042 | FS.ZER=  | 002000   | L.LEN=   | 000023 | Z.LMC=   | 000100   |
| CS.ABO= | 000100 | C.XTC=   | 000037 | G\$SLVL= | 000001   | L.MPF=   | 000022 | ZF.MAN=  | 020000   |
| CS.BRO= | 000002 | C.X25=   | 000036 | G\$STPP= | 000000   | L.NMST=  | 000020 | ZF.MFL=  | 000010   |
| CS.BUF= | 000200 | D\$BUG=  | 177514 | G\$STSS= | 000000   | L.NSTA=  | 000014 | ZF.MTM=  | 000400   |
| CS.CES= | 000002 | D\$ISK=  | 000000 | G\$STTK= | 000000   | L.DWNR=  | 000021 | ZF.MUX=  | 000040   |
| CS.CHN= | 000010 | D\$SLI=  | 000001 | G\$SWRD= | 000000   | L.UNT=   | 000015 | ZF.PSE=  | 002000   |
| CS.CMP= | 000200 | D\$SYNC= | 000000 | I\$SRAR= | 000000   | M\$SCRB= | 000124 | ZF.SLI=  | 010000   |
| CS.CDR= | 000400 | D\$SYNM= | 000000 | I\$SRDN= | 000000   | M\$SCRX= | 000000 | ZF.TIM=  | 000200   |
| CS.DEF= | 000004 | E\$XPR=  | 000000 | KISAR5=  | ***** GX | M\$SFCS= | 000000 | ZF.X3P=  | 000000   |
| CS.DEV= | 000002 | FC.CCP=  | 000020 | K\$CNT=  | 177546   | M\$SMGE= | 000000 | ZS.ASN=  | 100000   |
| CS.DIS= | 000040 | FC.CTL=  | 000006 | K\$CSR=  | 177546   | M\$SNET= | 000000 | ZS.BSY=  | 140000   |
| CS.ENA= | 0000   | FC.KCP=  | 000016 | K\$LDC=  | 000000   | M\$SOVR= | 000000 | Z.AVL=   | 000014   |

## Symbol table

|                  |                  |                |                      |                       |
|------------------|------------------|----------------|----------------------|-----------------------|
| ASSCHK= 000000   | CS.CES= 000002   | DSSBUG= 177514 | GSSISS= 000000       | SEMASK 000314RG 002   |
| ASSCPS= 000000   | CS.CHN= 000010   | DSSISK= 000000 | GSSITK= 000000       | SETIMR 000314RG 002   |
| ASSPRI= 000000   | CS.CMP= 000200   | DSSL11= 000001 | GSSWRD= 000000       | SEVDSC 000226RG 002   |
| ASSTRP= 000000   | CS.DCR= 000400   | DSSYNC= 000000 | ISSRAR= 000000       | SFILHD 000222RG 002   |
| CB.CCB= 000002   | CS.DEF= 000004   | DSSYNM= 000000 | ISSRDN= 000000       | SIMASK 000314RG 002   |
| CB.DDM= 000040   | CS.DEV= 000002   | ESSXPR= 000000 | KSSCNT= 177548       | SLDBAF 000032RG 002   |
| CB.DLC= 000020   | CS.DIS= 000040   | FC.CCP= 000020 | KSSCSR= 177548       | SLGCON 000300RG 002   |
| CB.RDB= 000004   | CS.ENA= 000001   | FC.CTL= 000006 | KSSLDC= 000000       | SLGDDB 000220RG 002   |
| CB.SDB= 000010   | CS.ENB= 000020   | FC.KCP= 000016 | KSSTPS= 000074       | SLGFNB 000246RG 002   |
| CB.SLI= 000100   | CS.ERR= 000000   | FC.KIL= 000004 | K.CSR 000002 G       | SLGMON 000274RG 002   |
| CB.XLB= 000001   | CS.FTL= 001000   | FC.MAN= 000024 | K.PRI 000000 G       | SLGPDV 000216RG 002   |
| CC.LLC= 000200   | CS.HCR= 000001   | FC.MLD= 000026 | K.VCT 000001 G       | SLGSTT 000224RG 002   |
| CE.ABO= 100362   | CS.HFE= 002000   | FC.PCT= 000030 | LD\$LP = 000000      | SLGUIC 000272RG 002   |
| CE.DAO= 100346   | CS.LST= 040000   | FC.PWR= 000022 | LSSASG= 000000       | SLLCTA 000004RG 002   |
| CE.DIS= 100368   | CS.MTL= 004000   | FC.PCE= 000002 | LSSDRV= 000000       | SLTMFC 000104RG 002   |
| CE.ERR= 100370   | CS.RNG= 000010   | FC.RCP= 000014 | LSSP11= 000001       | \$MAXOV 000214RG 002  |
| CE.ILN= 100350   | CS.ROV= 000004   | FC.TIM= 000010 | LSS11R= 000000       | \$NBIAS 000166RG 002  |
| CE.LTO= 100356   | CS.RSH= 010000   | FC.XCP= 000012 | MSSCRB= 000124       | \$NMCLH 000306RG 002  |
| CE.MOP= 100372   | CS.SHU= 000001   | FC.XME= 000000 | MSSCRX= 000000       | \$NMCL2 000310RG 002  |
| CE.NTE= 100361   | CS.SID= 000002   | FS.AST= 000000 | MSSFCs= 000000       | \$NMMLST 000204RG 002 |
| CE.RTE= 100376   | CS.STR= 000004   | FS.CIB= 002000 | MSSMGE= 000000       | \$NTLPT 000162RG 002  |
| CE.SRC= 100364   | CS.SUC= 000001   | FS.CRA= 001000 | MSSNET= 000000       | \$OBJHD 000206RG 002  |
| CE.STP= 100352   | CS.TMD= 020000   | FS.DJS= 013000 | MSSOVR= 000000       | \$PAD = 000377 G      |
| CE.TME= 100354   | CS.XUR= 000004   | FS.DVC= 001000 | NSSACC= 000001       | \$PADB 000332RG 002   |
| CE.TMO= 100374   | \$SSCKP= 000000  | FS.ENB= 012000 | NSSBUF= 000001       | \$PADBF 000340RG 002  |
| CE.UNS= 100344   | \$SSORE= 000400  | FS.EXI= 001000 | NSSLDV= 000001       | \$PADKL= 000012 G     |
| CF.CHN= 000001   | \$SSRSH= 177564  | FS.GET= 006000 | NSSMCP= 000001       | \$PADSH= 000002 G     |
| CF.DDM= 000002   | C.ADD 000034     | FS.HLT= 000000 | NSSMLL= 000001       | \$PAVL 000202RG 002   |
| CF.DYN= 000004   | C.BID 000003     | FS.INI= 000000 | NSSMOV= 000010       | \$PBIAS 000320RG 002  |
| CF.EIS= 000010   | C.BUF 000014     | FS.KIL= 000000 | NSSNCT= 000001       | \$PDVNM 000006RG 002  |
| CF.FOM= 000004   | C.BUF1 000014    | FS.LCL= 100000 | NSSPEM= 000001       | \$PDVTA 000000RG 002  |
| CF.FRK= 100000   | C.BUF2 000024    | FS.LTM= 001000 | PSSP45= 000000       | \$PSIPT 000126RG 002  |
| CF.HDR= 000020   | C.CNT 000020     | FS.MNT= 004000 | PSSWRD= 000000       | \$PUMR 000174RG 002   |
| CF.LB = 100000   | C.CNT1 000020    | FS.MSN= 014000 | QSSOPT= 000010       | \$PWRf1 000176RG 002  |
| CF.LIN= 000002   | C.CNT2 000030    | FS.REA= 001000 | RSSDER= 000000       | \$QBIA5 000170RG 002  |
| CF.LOG= 000020   | C.FLG 000022     | FS.RET= 000000 | RSSK11= 000001       | \$QSTRT 000172RG 002  |
| CF.MDM= 000001   | C.FLG1 000022    | FS.REZ= 003000 | RSSND= 000000        | \$RDBAF 000154RG 002  |
| CF.SOM= 000010   | C.FLG2 000032    | FS.RLB= 002000 | RSS11M= 000000       | \$RDBCT 000146RG 002  |
| CF.SYN= 000040   | C.FNC 000010     | FS.RNG= 011000 | SSSWRG= 000000       | \$RDBLH 000150RG 002  |
| CF.TIM= 000400   | C.LIN 000006     | FS.RST= 000000 | SSSYSZ= 007600       | \$RDBNM 000016RG 002  |
| CF.TRN= 000100   | C.LNK 000000     | FS.RTN= 001000 | TSSKMG= 000000       | \$RDBSZ 000020RG 002  |
| CM.CIR= 000002   | C.MOD 000011     | FS.SET= 005000 | TSSMIN= 000000       | \$RDBTH 000036RG 002  |
| CM.FMT= 100000   | C.NSP 000004     | FS.SFC= 005000 | VSSCTR= 001000       | \$RDQCT 000160RG 002  |
| CM.HRD= 000002   | C.PRO 000042     | FS.SFR= 006000 | XSSDBT= 000000       | \$RDRF 000156RG 002   |
| CM.LIN= 000000   | C.RSV 000002     | FS.SFS= 004000 | \$CCBAF 000030RG 002 | \$SDP 000144RG 002    |
| CM.LOC= 000001   | C.STA 000007     | FS.SPW= 004000 | \$CCBAL 000034RG 002 | \$SDE 000136RG 002    |
| CM.XLD= 000004   | C.STS 000012     | FS.STM= 000000 | \$CCBCT 000026RG 002 | \$SDBLH 000140RG 002  |
| CP.DCF= 000040   | C.URM 177776     | FS.STP= 002000 | \$CCBLH 000134RG 002 | \$SDBNM 000022RG 002  |
| CP.HDL= 000007   | C.XACP 000004    | FS.STR= 001000 | \$CCBNM 000012RG 002 | \$SSBSZ 000024RG 002  |
| CP.PS = 177400   | C.XID 000035     | FS.TRM= 003000 | \$CCBSZ 000014RG 002 | \$SHLST 000312RG 002  |
| CP.PSI= 000200   | C.XLEN 000044    | FS.WLB= 001000 | \$CEAVL 000002RG 002 | \$SLTMA 000002RG 002  |
| CP.XCF= 000100   | C.XPLI 000040    | FS.XKL= 002000 | \$CELFN 000304RG 002 | \$SLTNM 000010RG 002  |
| CP.2FR= 000030   | C.XPT 000034     | FS.XOF= 010000 | \$CMFRK 000046RG 002 | \$SNAPT 000130RG 002  |
| CRCOV = 000002 G | C.XSVF 000042    | FS.XON= 007000 | \$CMPDV 000040RG 002 | \$SPAR1 000314RG 002  |
| CS.ABO= 000100   | C.XTC 000037     | FS.ZER= 002000 | \$CXOPT 000164RG 002 | \$SPAR2 000320RG 002  |
| CS.BRO= 000002   | C.XZS 000036     | FSSVL= 000001  | \$DDFNC 000132RG 002 | \$SQRCM 000212RG 002  |
| CS.BUF= 000200   | DLCOV = 000010 G | GSSTPP= 000000 | \$DECP1 000124RG 002 | \$STMFC 000064RG 002  |



```

166
167 000034 012302 MOV (R3)+,R2 ; GET VIRTUAL ADDRESS OF BUFFER
168
169 .IFT ; M$$MGE
170 .IF NDF 1$$AS
171
172 000036 022702 140000 CMP #140000,R2 ; IS BUFFER MAPPED BY AND APR < 6?
173 000042 101002 BHI 20$; IF H1, YES
174 000044 162702 020000 SUB #20000,R2 ; MAP BUFFER THROUGH APR5
175
176 .IFF ; 1$$AS
177 CMP #60000,R2 ; IS BUFFER MAPPED BY AN APR 3 ?
178 BLOS 20$; IF LO OR SAME, NO - MUST BE APR 3
179 ADD #20000,R2 ; MAP BUFFER THROUGH APR3
180 .ENDC ; 1$$AS
181 000050 20$: .ENDC ; M$$MGE
182
183
184 000050 012300 MOV (R3)+,R0 ; GET LENGTH OF SEGMENT
185 000052 CALL STCR2 ; COMPUTE CRC ON SEGMENT
186 000056 J32713 000004 BIT #CF.EOM,(R3) ; LAST SEGMENT IN MESSAGE?
187 000062 001407 BEQ 30$; IF EQ, NO
188
189 000064 060002 ADD R0,R2 ; POINT TO END OF MESSAGE
190 000066 110122 MOVVB R1,(R2)+ ; STORE CRC FOLLOWING DATA
191 000070 000301 SWAB R1 ; ...
192 000072 110122 MOVVB R1,(R2)+ ; ...
193 000074 062743 000002 ADD #2,-(R3) ; UPDATE COUNT IN CCB
194 000100 005001 CLH R1 ; RE-INITIALISE CRC ACCUMULATOR
195
196 000102 011404 30$: MOV (R4),R4 ; GET NEXT CCB IN CHAIN
197 000104 001346 BNE 10$; AND COMPUTE IT'S CRC
198
199 000106 RESRG <R4,R1,R0> ; RESTORE REGISTERS
200
201 .IF DF M$$MGE
202 .IF NDF 1$$AS
203
204 000114 012677 177662 MOV (SP)+,@KSARS ; RESTORE MAPPING
205
206 .IFF ; 1$$AS
207
208 MOV (SP)+,@(SP)+ ; RESTORE MAPPING
209
210 .ENDC ; 1$$AS
211 .ENDC ; M$$MGE
212
213 000120 RETURN

```

\*\*wjd01\*\*

STCRC MACRO V05.03b Friday 28-Jun-85 18:21 Page 11-7  
Symbol table

Work file reads: 0  
Work file writes: 0  
Size of work file: 15190 Words ( 60 Pages)  
Size of core pool: 16552 Words ( 63 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:15.05

SY:STCRC.V2,[130,134]STCRC/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]STCRC

STCRC MACRO V05.03b Friday 28-Jun-85 18:22 Page 10  
 \$STCRC - CALCULATE CRC ON BLOCK OF DATA

```

215 .SBTTL $STCRC - CALCULATE CRC ON BLOCK OF DATA
216
217 :+
218 **-$STCRC-CALCULATE CRC ON BLOCK OF DATA
219 **-$STCR1-ALTERNATE ENTRY (R1 - PARTIAL CRC)
220 :
221 CALCULATE THE CRC ON A BLOCK OF DATA.
222 :
223 INPUTS:
224
225 R0 - # OF BYTES IN DATA BLOCK
226 R1 - PARTIAL CRC OR 0
227 R2 - POINTER TO DATA BLOCK
228 R4 - CCB DESCRIBING DATA BUFFER
229 :
230 OUTPUTS:
231 R1 - UPDATED CRC VALUE
232
233 000122 005001 $STCRC::CLR R1 ; INITIALISE CRC ACCUMULATOR
234
235 000124 $STCR1::
236 .IF DF M$$MGE
237 .IF NDF I$$AS
238
239 000124 017746 177652 MOV @KSAR5,-(SP) ; SAVE CURRENT MAPPING
240 000130 016477 000014 177644 MOV C.BUF(R4),@KSAR5 ; AND MAP TO BUFFER
241 000136 022702 140000 CMP #140000,R2 ; IS BUFFER MAPPED BY APR > 6?
242 000142 101002 BH 10$; IF H1, NO
243 000144 162702 020000 SUB #20000,R2 ; MAP BUFFER USING APR5
244
245 .IFF ; I$$AS
246
247 MOV #KP.AR3,-(SP) ; ASSUME KERNEL MODE
248 BIT #140000,PS.EXP ; KERNEL MODE ?
249 BEQ 5$; YES - BR
250 5$: MOV #UPAR0+6,(SP) ; NO - MUST BE USER MODE
251 MOV @C.BUF(R4),@2(SP) ; SAVE CURRENT MAPPING
252 MOV C.BUF(R4),@2(SP) ; MAP TO DATA BUFFER
253 CMP #60000,R2 ; IS BUFFER MAPPED BY APR 2?
254 BLOS 10$; IF LO OR SAME, NO
255 ADD #20000,R2 ; MAP BUFFER USING APR3
256
257 .ENDC ; I$$AS
258 10$: .ENDC ; M$$MGE
259
260 000150 CALL STCR2 ; CALCULATE CRC
261
262 .IF DF M$$MGE
263 .IF NDF I$$AS
264
265 000154 012677 177622 MOV (SP)+,@KSAR5 ; RESTORE MAPPING
266
267 .IFF
268
269 MOV (SP)+,@(SP)+ ; RESTORE MAPPING
270
271 .ENDC ; I$$AS

```

STCRCF CREATED BY MACRO ON 28-JUN-85 AT 18:22

PAGE 1 D 8

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                                       |
|---------|------------|--------------------------------------------------|
| BUFUMP  | = 172354   | #6-68                                            |
| CF.EQM  | = 000004   | 9-186                                            |
| CLRKG   | = 000020   | #7-96 7-98                                       |
| CMODE   | = 140000   | #6-68                                            |
| CRC16   | = 000001   | #7-93 7-99                                       |
| CTABL   | 000234 R   | 11-361 #12-436                                   |
| C.BUF   | 000014     | 9-149 10-239                                     |
| DDB     | = 000010   | #7-95 7-98 7-99                                  |
| F\$AST  | = 000001   | #4-2 11-351 12-453                               |
| I\$AS   | = *****    | 6-68 9-130 9-152 9-170 9-202 10-236 10-263       |
| KGCSR   | = 170700   | #7-92                                            |
| KGINIT  | = 000111   | #7-99                                            |
| KGLDBC  | = 000133   | #7-98                                            |
| KISARO  | = 172340   | #6-68                                            |
| KISAR5  | = ***** GX | 8-104                                            |
| KISAR6  | = 172354   | #6-68 8-105                                      |
| KSAR5   | 000002 R   | #8-104 9-132 9-154 9-204 10-238 10-239 10-265    |
| KSAR6   | 000004 R   | #8-105                                           |
| K\$SG11 | = *****    | 5-1 11-293 12-431                                |
| LRC16   | = 000003   | #7-94 7-98                                       |
| L\$SI1  | = *****    | 6-73                                             |
| MPAR    | = 172100   | #6-68                                            |
| MPCSR   | = 177746   | #6-68                                            |
| M\$MGE  | = 000000   | 6-68 6-73 9-129 9-151 9-201 10-235 10-262 11-359 |
| PIRQ    | = 177772   | #6-68                                            |
| PMODE   | = 030000   | #6-68                                            |
| PR0     | = 000000   | #6-68                                            |
| PR1     | = 000040   | #6-68                                            |
| PR2     | = 000100   | #6-68                                            |
| PR3     | = 000140   | #6-68                                            |
| PR4     | = 000200   | #6-68                                            |
| PR5     | = 000240   | #6-68                                            |
| PR6     | = 000300   | #6-68                                            |
| PR7     | = 000340   | #6-68                                            |
| PS      | = 177776   | #6-68                                            |
| P\$S34  | = *****    | 6-73                                             |
| P\$S40  | = *****    | 6-73                                             |
| P\$S45  | = *****    | 6-73                                             |
| P\$S70  | = *****    | 6-73                                             |
| R\$EIS  | = 000000   | #6-75 6-79 11-344 11-420                         |
| R\$MPL  | = *****    | 6-68 6-68                                        |
| R\$S11C | = *****    | 6-68                                             |
| SEN     | = 000100   | #7-97 7-98 7-99                                  |
| STCR2   | 000162 R   | 9-185 10-260 #11-291                             |
| SWR     | = 177570   | #6-68                                            |
| TPS     | = 177564   | #6-68                                            |
| UBMPR   | = 170200   | #6-68                                            |
| UISARO  | = 177640   | #6-68                                            |
| UISAR1  | = 177642   | #6-68                                            |
| \$LRCRC | 000006 RG  | #9-128                                           |
| \$STCP  | 000122 RG  | #10-232                                          |
| \$STCR1 | 000124 RG  | #10-234                                          |

STCRCK MACRO V05.03b Friday 28-Jun-85 18:22 Page 10-1  
\$STCRC - CALCULATE CRC ON BLOCK OF DATA

D 9

272  
273  
274 000160

.ENDC ; M\$\$MGE

RETURN

STCRCK      CREATED BY    MACRO    ON 28-JUN-85 AT 18:22      PAGE 2      D 10

SYMBOL CROSS REFERENCE

CREF    04.00

SYMBOL    VALUE                      REFERENCES

.BASEB    = 140000                      #6-68

AXBFR CREATED BY MACRO ON 28-JUN-85 AT 18:28

PAGE 2 D 11

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL | VALUE      | REFERENCES                            |
|--------|------------|---------------------------------------|
| L.FLG  | 000000     | #5-60                                 |
| L.KRBA | 000016     | #5-60                                 |
| L.LEN  | = 000022   | #5-60                                 |
| L.MPF  | 000022     | #5-60                                 |
| L.NMST | 000020     | #5-60                                 |
| L.NSTA | 000014     | #5-60                                 |
| L.OWNR | 000021     | #5-60                                 |
| L.UNT  | 000013     | #5-60                                 |
| MPAR   | = 172100   | #5-61                                 |
| MPCSR  | = 177746   | #5-61                                 |
| M\$MGE | = 000000   | 5-61                                  |
| N\$SLN | = *****    | 6-116                                 |
| PDSPL  | = ***** GX | 6-156                                 |
| PIRQ   | = 177772   | #5-61                                 |
| PMODE  | = 030000   | #5-61                                 |
| PR0    | = 000000   | #5-61 6-106 6-166 6-186               |
| PR1    | = 000040   | #5-61                                 |
| PR2    | = 000100   | #5-61                                 |
| PR3    | = 000140   | #5-61                                 |
| PR4    | = 000200   | #5-61                                 |
| PR5    | = 000240   | #5-61                                 |
| PR6    | = 000300   | #5-61                                 |
| PR7    | = 000340   | #5-61                                 |
| PS     | = 177776   | #5-61 6-96 *6-96 *6-106 *6-166 *6-186 |
| RDBG   | = ***** GX | 6-170                                 |
| RDBRT  | = ***** GX | 6-178                                 |
| RDBSZ  | = ***** GX | 6-92 6-93                             |
| RDOCT  | = ***** GX | 6-97 6-132                            |
| RDOGL  | = ***** GX | 6-110 6-112 6-113                     |
| R\$MPL | = *****    | 5-61 5-61 6-78 6-134 6-190 6-202      |
| R\$11D | = *****    | 5-61 6-100 6-160 6-180                |
| SF.ACT | = 000200   | #5-60                                 |
| SF.ENA | = 000100   | #5-60                                 |
| SF.LPB | = 000004   | #5-60                                 |
| SF.MFL | = 000040   | #5-60                                 |
| SF.PAC | = 000020   | #5-60                                 |
| SF.REA | = 000010   | #5-60                                 |
| SF.SER | = 000001   | #5-60                                 |
| SF.SVC | = 000002   | #5-60                                 |
| SF.UNL | = 000040   | #5-60                                 |
| SLTMA  | = ***** GX | 6-124                                 |
| SLTNM  | = ***** GX | 6-112                                 |
| SWR    | = 177570   | #5-61                                 |
| S.COST | = 000001   | #5-60                                 |
| S.FLG  | = 000000   | #5-60                                 |
| S.LEN  | = 000004   | #5-60                                 |
| S.NMST | = 000002   | #5-60                                 |
| S.OWNR | = 000003   | #5-60                                 |
| TPS    | = 177564   | #5-61                                 |
| UBMPR  | = 170200   | #5-61                                 |
| UISAR0 | = 177640   | #5-61                                 |
| UISAR1 | = 177642   | #5-61                                 |

AXDSP - AUXILLIARY PROCESS DISP MACRO V05.03b Tuesday 03-Sep-85 10:58 <sup>D 12</sup>  
Table of contents

9- 254 NETWORK MANAGEMENT COUNTER COMPLETION



AXDSP CREATED BY MACRO ON 3-SEP-85 AT 10:58

PAGE 1 D 13

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL | VALUE      | REFERENCES   |
|--------|------------|--------------|
| CB.CCB | = 000002   | 9-306        |
| CCBRET | 000030 R   | 6-106 #6-120 |
| CCBRT  | = ***** GX | 6-120 9-308  |
| CS.LST | = 040000   | 9-303        |
| C.ADD  | 000034     | 9-270        |
| C.BID  | 000003     | 9-306        |
| C.CNT2 | 000030     | 9-274 9-286  |
| C.STS  | 000012     | *9-303       |
| DUMMY  | 000026 R   | 6-100 #6-119 |
| EXRQN  | = ***** GX | 9-304        |
| ISSAS  | = *****    | 4-1 6-115    |
| LF.ACT | = 100000   | #5-77        |
| LF.BRO | = 000400   | #5-77 9-277  |
| LF.BWT | = 000007   | #5-77        |
| LF.ENA | = 002000   | #5-77        |
| LF.LPB | = 001000   | #5-77        |
| LF.MDC | = 000100   | #5-77        |
| LF.MFL | = 004000   | #5-77        |
| LF.MTP | = 000020   | #5-77 9-279  |
| LF.PAC | = 000200   | #5-77        |
| LF.RDY | = 040000   | #5-77        |
| LF.REA | = 010000   | #5-77        |
| LF.SER | = 000040   | #5-77 9-281  |
| LF.TIM | = 000010   | #5-77        |
| LF.UNL | = 020000   | #5-77        |
| LF.X2P | = 000000   | #5-77        |
| LN.CLO | = 000000   | #5-77        |
| LN.DUM | = 000005   | #5-77        |
| LN.LOA | = 000004   | #5-77        |
| LN.LOO | = 000003   | #5-77        |
| LN.OAU | = 000003   | #5-77        |
| LN.OFF | = 000001   | #5-77        |
| LN.ON  | = 000000   | #5-77        |
| LN.OOP | = 000004   | #5-77        |
| LN.OPE | = 000001   | #5-77        |
| LN.REF | = 000002   | #5-77        |
| LN.SER | = 000002   | #5-77        |
| LN.STA | = 000017   | #5-77        |
| LN.SUB | = 000360   | #5-77        |
| LN.TRI | = 000006   | #5-77        |
| L.COST | 000015     | #5-77        |
| L.CTL  | 000012     | #5-77        |
| L.CVA  | 177776     | #5-77        |
| L.DDM  | 000002     | #5-77        |
| L.DDS  | 000004     | #5-77        |
| L.DLC  | 000003     | #5-77        |
| L.DLM  | 000006     | #5-77        |
| L.DLS  | 000010     | #5-77        |
| L.FLG  | 000000     | #5-77        |
| L.KRBA | 000016     | #5-77        |
| L.LEN  | = 000022   | #5-77 9-285  |
| L.MPF  | 000022     | #5-77        |

```

254 .SBTTL NETWORK MANAGEMENT COUNTER COMPLETION
255
256 ***-NMCMP-NETWORK MANAGEMENT COUNTER COMPLETION
257
258 THIS ROUTINE IS ENTERED WHEN A NETWORK MANAGEMENT REQUEST FOR COUNTERS
259 HAS BEEN COMPLETED BY A LOWER LEVEL PROCESS.
260 THE ROUTINE ALSO REQUESTS/UNSTOPS NETACP TO PROCESS
261 REQUESTS FOR LINE WATCHER
262
263 INPUTS:
264 R4 = ADDRESS OF COUNTER CCB
265
266
267 000306 054134 003310 NETACP: .RAD50 /NETACP/
268
269 000312 010403 NMCMP: MOV R4,R3 ; Copy the ccb address
270 000314 062703 ADD #C.ADD,R3 ; Point to the task name
271 000320 005713 TST (R3) ; Is it for NETACP ?
272 000322 001041 BNE 40$; If NE, no
273
274 000324 116403 000030 MOV B C.CNT2(R4),R3 ; Copy SLN
275 000330 006303 R3 ; Form word index
276 000332 067703 000000G ADD @SLTMA,R3 ; Point to line table address
277 000336 032773 000400 000000 BIT #LF.BRO,@(R3) ; Is it a broadcast channel?
278 000344 001026 BNE 30$; If NE, yes - EPM already filtered it.
279 000346 032773 000020 000000 BIT #LF.MTP,@(R3) ; Is it multipoint?
280 000354 001005 BNE 5$; If NE, yes - check station table
281 000356 032773 000040 000000 BIT #LF.SER,@(R3) ; Is service disabled ?
282 000364 001416 BEQ 30$; If EQ, no - allow request
283 000366 000437 BR 60$; Else, toss the request
284 000370 011303 5$: MOV (R3),R3 ; Get to the end of the line table
285 000372 062703 000022 ADD #L.LEN,R3 ; ...
286 000376 116401 000031 MOV B C.CNT2+1(R4),R1 ; Get the station number
287 000402 001404 BEQ 20$; ...
288 000404 062703 000004 10$: ADD #S.LEN,R3 ; Move to the correct station table
289 000410 SOB R1,10$; ...
290 000414 032713 000001 20$: BIT #SF.SER,(R3) ; Is service disabled ?
291 000420 001022 BNE 60$; If NE, yes - toss the request
292
293 000422 012703 000306' 30$: MUV #NETACP,R3 ; Else start up NETACP
294 000426 40$: CALL @SRSTD ; Scan STD for task's TCB
295 000432 103415 BCS 60$; If CS, not there !!
296
297 000434 017703 000000G MOV @NMCL2,R3 ; Get address of net man listhead
298 000440 010413 MOV R4,(R3) ; Add CCB to end of head pointer
299 000442 010477 000000G 50$: MOV R4,@NMCL2 ; Update the tail pointer
300 000446 010403 MOV R4,R3 ; Copy possible 'last' buffer address
301 000450 011404 MOV (R4),R4 ; Else, get to the end of this chain
302 000452 001373 BNE 50$; If NE, get next buffer
303 000454 052763 040000 000012 BIS #CS.LST,C.STS(R3) ; Else, set end of chain flag
304 000462 CALLR @EXRQW ; Request the task (unstop or request)
305
306 000466 122764 000002 000003 60$: CMPB #CB.CCB,C.BID(R4) ; Is it a CCB ?
307 000474 001002 BNE 70$; If NE, no
308 000476 CALLR @CCBRT ; Return the CCB
309 000502 70$: SAVRG <(R4)> ; Save the next in the chain
310 000504 CALL @RDBRT ; Return this one

```

122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178

```
.IF DF P$$$RFL
.SBTTL POWERFAIL RECOVERY ROUTINE

+
**PWFAIL-POWERFAIL RECOVERY ROUTINE

THIS ROUTINE IS INVOKED ONCE PER SECOND BY THE TIMER SERVICE
CODE. IF THE POWERFAIL RECOVERY FLAG IS SET, WE WILL SCAN THE
SYSTEM LINE TABLE FOR 'ACTIVE' LINES (I.E. LINES WHICH HAVE
BOTH DLC AND DDM PROCESSES LOADED) AND ASYNCHRONOUSLY QUEUE
A CONTROL COMPLETION TO THE LLC LEVEL INDICATING THAT THE
LINK HAS BEEN DISCONNECTED.

-

PWFAIL: MOV @PWRFL,R1 ; GET # OF LINES REMAINING TO BE POWERFAILED
 BLE 100$; IF NONE ... NO RECOVERY UNDERWAY
 DEC R1 ; CONVERT TO SYSTEM LINE #
 MOV R1,R3 ; SAVE FOR LATER CALL TO $ASCMP

 .IF DF N$$$1LN

 MOV @SLTMA,R1 ; GET ADDRESS OF SYSTEM LINE TABLE
 MOV (R1),R1 ; ...

 .IFF

 ASL R1 ; FORM WORD INDEX
 ADD @SLTMA,R1 ; POINT INTO SYSTEM LINE MAPPING TABLE
 MOV (R1),R1 ; GET ADDRESS OF SYSTEM LINE TABLE

 .ENDC

 BIT #LF.ACT,(R1) ; IS THIS LINE 'ACTIVE'?
 BEQ 20$; NO ... NO RECOVERY REQUIRED
 CLR R2 ; ASSUME LINE IS NOT MULTIPOINT
 TSTB L,NSTA(R1) ; IS THIS LINE MULTIPOINT?
 BEQ 10$; IF EQ, NO
 MOVB TRIB+1,R2 ; GET TRIBUTARY NUMBER TO CHECK
 ASL R2 ; FORM DOUBLE WORD OFFSET
 ADD R1,R2 ; ...
 BITB #SF.ACT,L.MPF(R2) ; IS THE TRIBUTARY ACTIVE?
 BEQ 15$; NO ... NO RECOVERY REQUIRED
 MOV TRIB,R2 ; YES ... GET TRIBUTARY ADDRESS
 BISB R3,R2 ; SET SYSTEM LINE NUMBER
 MOV #CE.DIS,R3 ; YES ... SET UP ERROR CODE
 CALL @ASCMP ; PERFORM ASYNCHRONOUS COMPLETION
 BCS 100$; TRY LATER ON RESOURCE ALLOCATION FAILURE
 INCB TRIB+1 ; UPDATE TRIBUTARY ADDRESS
 CMPB TRIB+1,L.NSTA(R1) ; HAVE WE CHECKED ALL TRIBUTARIES ON THIS LINE?
 BLO PWFAIL ; IF LO, NO
 CLR TRIB ; RESET TRIBUTARY ADDRESS
 DEC @PWRFL ; ONE LESS LINE TO RECOVER
 BNE PWFAIL ; LOOP TILL ALL DONE

 10$:
 15$:
 20$:
 100$:

 .IF DF X$$$MDC
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

```
.IF DF R$$$1D!$$$S
.IF DF X$$$MDC
.TITLE AXDSPP - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSP - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.ENDC
.IFF
.IF DF X$$$MDC
.IF DF P$$$RFL
.TITLE AXDSPB - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSPP - AUXILLIARY PROCESS DISPATCH
.ENDC
.IFF
.IF DF P$$$RFL
.TITLE AXDSPP - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSP - AUXILLIARY PROCESS DISPATCH
.ENDC
.ENDC
.ENDC
.IDENT /V05.01/
```

```
:
: COPYRIGHT (C) 1978,1979,1980 1983, 1985 BY
: DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
```

```
:
: THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
: ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
: INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
: COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
: OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
: TRANSFERRED.
```

```
:
: THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
: AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
: CORPORATION.
```

```
:
: DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
: SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
```

```
:
: MODULE DESCRIPTION
```

```
:
: AUXILARY PROCESS DISPATCH TABLE
```

```
:
: DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
```

```
:
: IDENT HISTORY:
```

```
:
: 1.00 10-FEB-78
: VERSION 2.0 RELEASE
:
: 2.00 14-DEC-79
```

```
58 ; MACRO LIBRARY CALLS
59 ;
60 .MCALL INHIB$,ENABL$,SAVRG,RESRG
61 .MCALL CCBDF$,PDVDF$,CLKDF$
62 .MCALL CALLR ; AVOID SYSTEM DEPENDENCY
63 CCBDF$; DEFINE THE CCB OFFSETS
64 PDVDF$; DEFINE THE PDV OFFSETS
65 CLKDF$; DEFINE CLOCK BLOCK OFFSETS
66
```

537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556

000001

```
.SBTTL $MPSAV - BYPASS CACHE WITH SAVE AND RESTORE
;+
; **-$MPSAV-BYPASS CACHE WITH SAVE AND RESTORE
; THIS ROUTINE WILL SAVE THE CURRENT STATE OF THE CACHE AND BYPASS
; IT BEFORE CO-CALLING THE CALLER. ON RETURN IT WILL RESTORE THE
; INITIAL STATE OF THE CACHE.
;-
$MPSAV::CACHE$ SAVE ; SAVE CURRENT STATE OF CACHE
CALL @2(SP) ; CO-CALL THE CALLER
MOV (SP)+,2(SP) ; OVERWRITE RETURN LINK
CACHE$ UNSAVE ; RESTORE THE CACHE
RETURN
.ENDC ; DF M$PRO
;
.END
```

CETIM MACRO VC5.03b Friday 28-Jun-85 18:21 Page 6-2  
Symbol table

|       |        |         |        |       |        |          |          |            |          |
|-------|--------|---------|--------|-------|--------|----------|----------|------------|----------|
| Z.DAT | 000016 | Z.LEN = | 000016 | Z.NAM | 000004 | \$DSPTM  | 000052RG | \$PDVIA=   | ***** GX |
| Z.DSP | 000000 | Z.LLN   | 000006 | Z.PCB | 000012 | \$LTMFC= | ***** GX | \$ISTIM    | 000000RC |
| Z.FLG | 000010 | Z.MAP   | 000020 | Z.SCH | 000007 | \$PDSPL= | ***** GX | .\$\$\$\$= | 000034   |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
C00074 001 (RW,I,LCL,REL,CON)

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 9  
Work file writes: 12  
Size of work file: 17471 Words ( 39 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:12.04

SY:CETIM.V2,[130,134]CETIM/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]ETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]CETIM

CEXCM - CEX COMMON DATABASE  
Symbol table

MACRO V05.03b Friday 28-Jun-85 18:21 <sup>E 4</sup> Page 5-6

|          |          |     |         |          |     |         |          |     |         |          |     |            |          |     |
|----------|----------|-----|---------|----------|-----|---------|----------|-----|---------|----------|-----|------------|----------|-----|
| \$SYNB   | 000324RG | 002 | \$ICK50 | 000210RG | 002 | \$T100C | 000066RG | 002 | \$T50Q  | 000060RG | 002 | \$ZTIME    | 000042RG | 002 |
| \$SYNBF  | 000352RG | 002 | \$IK100 | 000210RG | 002 | \$T100Q | 000060RG | 002 | \$XAVL  | 000202RG | 002 | \$ZTIM2    | 000044RG | 002 |
| \$SYNC = | 000226 G |     | \$TISCL | 000106RG | 002 | \$T50CL | 000066RG | 002 | \$XBIAS | 000200RG | 002 | .\$\$\$\$. | 000034   |     |
| \$SYNCT= | 000010 G |     |         |          |     |         |          |     |         |          |     |            |          |     |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000000 001 (RW,I,LCL,REL,CON)  
CEXCM 000362 002 (RW,I,LCL,REL,CON)  
\$\$\$POL 000006 003 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 14625 Words ( 58 Pages)  
Size of core pool: 15496 Words ( 59 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:11.47

SY:CEXCM.V2,[130,134]CEXCM/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]CEXCM



```

215 .SBTTL $STCRC - CALCULATE CRC ON BLOCK OF DATA
216
217 +
218 **-$STCRC-CALCULATE CRC ON BLOCK OF DATA
219 **-$STCR1-ALTERNATE ENTRY (R1 - PARTIAL CRC)
220
221 CALCULATE THE CRC ON A BLOCK OF DATA.
222
223 INPUTS:
224
225 R0 - # OF BYTES IN DATA BLOCK
226 R1 - PARTIAL CRC OR 0
227 R2 - POINTER TO DATA BLOCK
228 R4 - CCB DESCRIBING DATA BUFFER
229
230 OUTPUTS:
231 R1 - UPDATED CRC VALUE
232
233 000122 005001 $STCRC::CLR R1 ; INITIALISE CRC ACCUMULATOR
234
235 000124 $STCR1::
236 .IF DF M$$MGE
237 .IF NDF I$$AS
238
239 000124 017746 177652 MOV @KSAR5,-(SP) ; SAVE CURRENT MAPPING
240 000130 016477 000014 177644 MOV C.BUF(R4),@KSAR5 ; AND MAP TO BUFFER
241 000136 022702 140000 CMP #140000,R2 ; IS BUFFER MAPPED BY APR > 6?
242 000142 101002 BHI 10$; IF HI, NO
243 000144 162702 020000 SUB #20000,R2 ; MAP BUFFER USING APR5
244
245 .IFF ; I$$AS
246
247 MOV #KP,AR3,-(SP) ; ASSUME KERNEL MODE
248 BIT #140000,PS.EXP ; KERNEL MODE ?
249 BEQ 5$; YES - BR
250 MOV #UPAR0+6,(SP) ; NO - MUST BE USER MODE
251 5$: MOV @C(SP),-(SP) ; SAVE CURRENT MAPPING
252 MOV C.BUF(R4),@2(SP) ; MAP TO DATA BUFFER
253 CMP #60000,R2 ; IS BUFFER MAPPED BY APR 2?
254 BLOS 10$; IF LO OR SAME, NO
255 ADD #20000,R2 ; MAP BUFFER USING APR3
256
257 .ENDC ; I$$AS
258
259 000150 10$: .ENDC ; M$$MGE
260
261 000150 CALL STCR2 ; CALCULATE CRC
262
263 .IF DF M$$MGE
264 .IF NDF I$$AS
265
266 000154 012677 177622 MOV (SP)+,@KSAR5 ; RESTORE MAPPING
267
268 .IFF
269
270 MOV (SP)+,@(SP)+ ; RESTORE MAPPING
271 .ENDC ; I$$AS

```

STCRC CREATED BY MACRO ON 28-JUN-85 AT 18:22

PAGE 1 E 6

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                                            |
|---------|------------|-------------------------------------------------------|
| BUFUMP  | = 172354   | #5-68                                                 |
| CF.EQM  | = 000004   | 8-186                                                 |
| CLRKG   | = 000020   | #6-96 6-98                                            |
| CMODE   | = 140000   | #5-68                                                 |
| CRC16   | = 000001   | #6-93 6-99                                            |
| CTABL   | 000256 R   | 10-385 10-401 #11-436                                 |
| C.BUF   | 000014     | 8-149 9-239                                           |
| DDB     | = 000010   | #6-95 6-98 6-99                                       |
| F\$SAST | = *****    | 4-4 10-351 11-453                                     |
| I\$SAS  | = *****    | 5-68 8-130 8-152 8-170 8-202 9-236 9-263              |
| KGCSR   | = 170700   | #6-92                                                 |
| KGINIT  | = 000111   | #6-99                                                 |
| KGLDBC  | = 000133   | #6-98                                                 |
| KISAR0  | = 172340   | #5-68                                                 |
| KISAR5  | = ***** GA | 7-104                                                 |
| KISAR6  | = 172354   | #5-68 7-105                                           |
| KSARS   | 000002 R   | #7-104 8-132 8-154 8-204 9-238 9-239 9-265            |
| KSAR4   | 000004 R   | #7-105                                                |
| K\$SG11 | = *****    | 4-1 10-293 11-431                                     |
| LRC16   | = 000003   | #6-94 6-98                                            |
| L\$S11  | = *****    | 5-73                                                  |
| MPAR    | = 172100   | #5-68                                                 |
| MPCSR   | = 177746   | #5-68                                                 |
| M\$MGE  | = 000000   | 5-68 5-73 8-129 8-151 8-201 9-235 9-262 10-383 10-399 |
| PIR0    | = 177772   | #5-68                                                 |
| PMODE   | = 030000   | #5-68                                                 |
| PR0     | = 000000   | #5-68                                                 |
| PR1     | = 000040   | #5-68                                                 |
| PR2     | = 000100   | #5-68                                                 |
| PR3     | = 000140   | #5-68                                                 |
| PR4     | = 000200   | #5-68                                                 |
| PR5     | = 000240   | #5-68                                                 |
| PR6     | = 000300   | #5-68                                                 |
| PR7     | = 000340   | #5-68                                                 |
| PS      | = 177776   | #5-68                                                 |
| P\$S34  | = *****    | 5-73                                                  |
| P\$S40  | = *****    | 5-73                                                  |
| P\$S45  | = *****    | 5-73                                                  |
| P\$S70  | = *****    | 5-73                                                  |
| R\$SEIS | = 000000   | #5-75 5-79 10-344 10-420                              |
| R\$SMPL | = *****    | 5-68 5-68                                             |
| R\$S11D | = *****    | 5-68                                                  |
| SEN     | = 000100   | #6-97 6-98 6-99                                       |
| STCR2   | 000162 R   | 8-185 9-260 #10-291                                   |
| SWR     | = 177570   | #5-68                                                 |
| TPS     | = 177564   | #5-68                                                 |
| UBMPR   | = 170200   | #5-68                                                 |
| UISAR0  | = 177640   | #5-68                                                 |
| UISAR1  | = 177642   | #5-68                                                 |
| \$CLCRC | 000006 RG  | #8-128                                                |
| \$STCRC | 000122 RG  | #9-232                                                |
| \$STCR1 | 000124 RG  | #9-234                                                |

STCRF MACRO V05.03b Friday 28-Jun-85 18:22 Page 10-1  
\$STCRC - CALCULATE CRC ON BLOCK OF DATA

E 7

272  
273  
274 000160

.ENDC ; M\$\$MGE  
RETURN

STCRF      CREATED BY MACRO ON 28-JUN-85 AT 18:22

PAGE 2      E 8

SYMBOL CROSS REFERENCE

CREF      04.00

| SYMBOL  | VALUE      | REFERENCES        |
|---------|------------|-------------------|
| \$STCVL | = 000002 G | #8-107            |
| \$STCVT | 000000 RG  | #8-103      8-107 |
| .BASEB  | = 140000   | #6-68             |

```

276 .SBTTL STCR2 - CALCULATE CRC
277
278 *--STCR2-CALCULATE CRC FOR A BLOCK OF DATA
279
280 CALCULATE THE CRC-16 FOR A BLOCK OF DATA USING THE KG-11 (IF AVAILABLE)
281 OR SOFTWARE.
282
283 INPUTS:
284 R0 - # OF BYTES IN DATA BLOCK
285 R1 - INITIAL CRC VALUE
286 R2 - POINTER TO DATA BLOCK
287
288 OUTPUTS:
289 R1 - UPDATED CRC VALUE
290
291 000162 STCR2: SAVRG <R0,R2,R4,R5> ; SAVE SOME REGISTERS
292
293 .IF DF K$G11
294
295 000172 012704 170700 MOV #K.GCSR,R4 ; POINT TO KG-11 CSR
296 000176 012705 170704 MOV #K.GCSR+4,R5 ; POINT TO KG-11 DATA REGISTER
297 000202 012446 MOV (R4)+,-(SP) ; SAVE CURRENT STATUS OF KG-11
298 000204 011446 MOV (R4),-(SP) ; SINCE IT IS SHARED
299 000206 012744 000133 MOV #KGLDBC,-(R4) ; SET UP TO LOAD NEW BCC
300
301 000212 105714 100376 10$: TSTB (R4) ; WAIT FOR COMPLETION
302 000214 100376 BPL 10$; (ONLY ON FASTER PROCESSORS)
303
304 000216 010115 MOV R1,(R5) ; LOAD NEW CRC
305 000220 012714 000111 MOV #KGINIT,(R4) ; AND INIT FOR CRC-16
306
307 000224 105714 20$: TSTB (R4) ; WAIT FOR COMPLETION
308 000226 100376 BPL 20$; ...
309
310 000230 032702 000001 BIT #1,R2 ; DOES BUFFER START ON ODD BOUNDARY?
311 000234 001411 BEQ 40$; IF EQ, NO
312 000236 042714 000010 BIC #DDB,(R4) ; SET KG-11 TO BYTE MODE
313 000242 112215 MOVB (R2)+,(R5) ; AND DO CRC ON BYTE
314 000244 105714 35$: TSTB (R4) ; Wait for completion
315 000246 100376 BPL 35$; ...
316 000250 052714 000010 BIS #DDB,(R4) ; RESET WORD MODE
317 000254 005300 DEC R0 ; REDUCE BYTE COUNT
318 000256 003412 BLE 80$; IF LE, ALL DONE
319
320 000260 006200 40$: ASR R0 ; CONVERT TO WORD COUNT
321 000262 006046 ROR -(SP) ; SAVE ODD BYTE COUNT ON STACK
322
323 000264 005300 50$: DEC R0 ; ANY MORE WORDS TO GO?
324 000266 002404 BLT 70$; IF LT, NO
325 000270 012215 MOV (R2)+,(R5) ; DO CRC ON WORD
326
327 000272 105714 60$: TSTB (R4) ; WAIT FOR COMPLETION
328 000274 100773 BMI 50$; IF MI, ALL DONE
329 000276 000775 BR 60$; CONTINUE WAITING
330
331 000300 005726 70$: TST (SP)+ ; CHECK FOR ODD BYTE COUNT
332 000302 100755 BMI 30$; IF MI, COMPUTE CRC ON LAST BYTE

```

STCRCK      CREATED BY    MACRO    ON 28-JUN-85 AT 18:22

PAGE 3      E 10

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME

REFERENCES

|         |       |        |        |
|---------|-------|--------|--------|
| CALL    | 9-185 | 10-260 |        |
| CCBDF\$ | #6-65 | 6-67   |        |
| NHWDF\$ | #6-65 | 6-68   |        |
| RESRG   | #6-65 | 9-199  | 11-428 |
| RETURN  | 9-213 | 10-274 | 11-429 |
| SAVRG   | #6-65 | 9-145  | 11-291 |

AXBFR      CREATED BY    MACRO    ON 28-JUN-85 AT 18:28

PAGE 3    E 11

SYMBOL CROSS REFERENCE

CREF    04.00

SYMBOL    VALUE                      REFERENCES

\$BFRIN    000000 RG                    #6-77  
.BASEB    = 140000                    #5-61

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

```
.IF DF RSS11D!$$$AS
.IF DF X$$$MDC
.TITLE AXDSPM - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSP - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.ENDC
.IFF
.IF DF X$$$MDC
.IF DF P$$$RFL
.TITLE AXDSPB - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSPM - AUXILLIARY PROCESS DISPATCH
.ENDC
.IFF
.IF DF P$$$RFL
.TITLE AXDSPP - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSP - AUXILLIARY PROCESS DISPATCH
.ENDC
.ENDC
.ENDC
.IDENT /V05.01/
```

COPYRIGHT (C) 1978,1979,1980, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

#### MODULE DESCRIPTION

AUXILARY PROCESS DISPATCH TABLE

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

#### IDENT HISTORY:

1.00 10-FEB-78  
VERSION 2.0 RELEASE  
2.00 14-DEC-79



AXDSP CREATED BY MACRO ON 3-SEP-85 AT 10:58

PAGE 2 E 13

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES            |
|---------|------------|-----------------------|
| L.NMST  | 000020     | #5-77                 |
| L.NSTA  | 000014     | #5-77                 |
| L.OWNR  | 000021     | #5-77                 |
| L.UNT   | 000013     | #5-77                 |
| NETACP  | 000034 R   | #9-267 9-293          |
| NMCL2   | = ***** GX | 9-297 9-299           |
| NMCMF   | 000040 R   | 6-112 #9-269          |
| P\$RFL  | = *****    | 4-15 6-97 6-107 7-122 |
| RDBRT   | = ***** GX | 9-310                 |
| R\$11D  | = *****    | 4-1 6-115             |
| SF.ACT  | = 000200   | #5-77                 |
| SF.ENA  | = 000100   | #5-77                 |
| SF.LPB  | = 000004   | #5-77                 |
| SF.MFL  | = 000040   | #5-77                 |
| SF.PAC  | = 000020   | #5-77                 |
| SF.REA  | = 000010   | #5-77                 |
| SF.SER  | = 000001   | #5-77 9-290           |
| SF.SVC  | = 000002   | #5-77                 |
| SF.UNL  | = 000040   | #5-77                 |
| SLTMA   | = ***** GX | 9-276                 |
| SRSTD   | = ***** GX | 9-294                 |
| S.COST  | 000001     | #5-77                 |
| S.FLG   | 000000     | #5-77                 |
| S.LEN   | 000004     | #5-77 9-288           |
| S.NMST  | 000002     | #5-77                 |
| S.OWNR  | 000003     | #5-77                 |
| X\$MDC  | = *****    | 4-8 6-88 6-115        |
| \$AUXTB | 000000 RG  | #6-84                 |
| \$BFRTN | = ***** GX | 6-86                  |
| \$MDCIN | 000026 RG  | #6-116                |

AXDSPB - AUXILIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 10:58 Page 10-1  
NETWORK MANAGEMENT COUNTER COMPLETION

311 000510  
312 000512 001373  
313 000514  
314  
315 000001

RESRG <R4> ; Recover the next buffer  
BNE 70\$ ; If NE, get next buffer  
RETURN ; Else, return  
.END

```
179 CALLR MDMSCN ; MODEM CONTROL SCAN ROUTINE
180 .IFF
181 RETURN
182 .ENDC
183
184 ;+
185 ; LOCAL STORAGE FOR TRIBUTARY ADDRESS DURING POWERFAIL
186 ; -
187
188 TRIB: .BLKW
```

|    |   |                                                                   |
|----|---|-------------------------------------------------------------------|
| 58 | : | DECNET-11M/S V3.0                                                 |
| 59 | : | DECNET-11M-PLUS V1.0                                              |
| 60 | : |                                                                   |
| 61 | : | 4.00 07-NOV-83                                                    |
| 62 | : | DECNET-11M V4.0                                                   |
| 63 | : | DECNET-11M-PLUS V2.0                                              |
| 64 | : |                                                                   |
| 65 | : | 5.00 22-JUL-85                                                    |
| 66 | : | DECnet-11M/S V4.2                                                 |
| 67 | : | DECnet-11M-Plus V3.0                                              |
| 68 | : | DECnet-Micro/RSX V1.0                                             |
| 69 | : |                                                                   |
| 70 | : | 5.01 09-Aug-85                                                    |
| 71 | : | Add logic to bypass service disabled check on broadcast channels. |
| 72 | : | This change requires EPMMAI ident V5.01.                          |
| 73 | : |                                                                   |

68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99

.SBTTL \$CMQIN - QUEUE A CHAIN OF CCBS TO A LIST

\*\*\*\$CMQIN - QUEUE A CHAIN OF CCBS TO A LIST

THIS SUBROUTINE IS CALLED TO ADD A CCB OR A CHAIN OF CCBS TO  
 THE END OF A SINGLE LINKED LIST.

INPUTS:

R3 = ADDRESS OF TWO WORD LISTHEAD  
 R4 = ADDRESS OF FIRST CCB IN CHAIN

OUTPUTS:

THE CCB IS QUEUED TO THE END OF THE LIST

REGISTERS MODIFIED:

NONE

```

$CMQIN::SAVRG <R5,R4> ; SAVE REGISTERS
10$: MOV R4,R5 ; COPY CCB ADDRESS
 BIC #CS.LST,C.STS(R5) ; CLEAR LAST CCB IN CHAIN BIT
 MOV (R5),R4 ; GET ADDRESS OF NEXT CCB IN CHAIN
 BNE 10$; LOOP TILL END OF CHAIN
 BIS #CS.LST,C.STS(R5) ; SET LAST CCB IN CHAIN BIT
 MOV (SP),@2(R3) ; LINK CHAIN TO END OF QUEUE
 MOV R5,2(R3) ; UPDATE LISTHEAD POINTER TO LAST CCB IN QUEUE
 RESRG <R4,R5> ; RESTORE REGISTERS
 RETURN ; RETURN

```

```

000000 010405
000004 042765 040000 000012
000014 011504
000016 001372
000020 052765 040000 000012
000026 011673 000002
000032 010563 000002
000036
000042

```

|                |                |                |                   |                    |
|----------------|----------------|----------------|-------------------|--------------------|
| ASSCHK= 000000 | CS.DIS= 000040 | C.SYTK= 000010 | FS.SET= 005000    | RSS11M= 000000     |
| ASSCPS= 000000 | CS.ENA= 000001 | C.TCB 000004   | FS.SFC= 005000    | SS\$WRG= 000000    |
| ASSPRI= 000000 | CS.ENB= 000020 | C.TIM 000006   | FS.SFR= 006000    | SS\$YSZ= 007600    |
| ASSTRP= 000000 | CS.ERR= 100000 | C.UIC 000016   | FS.SFS= 004000    | TSSKMG= 000000     |
| CB.CCB= 000002 | CS.FTL= 001000 | C.URM 177776   | FS.SPW= 040000    | TSSMIN= 000000     |
| CB.DDM= 000640 | CS.HCR= 000001 | C.XACP 000004  | FS.STM= 000000    | VSSCTR= 001000     |
| CB.DLC= 000020 | CS.HFE= 002000 | C.XID 000035   | FS.STP= 002000    | XSSDBT= 000000     |
| CB.RDB= 000004 | CS.LST= 040000 | C.XLEN 000044  | FS.STR= 001000    | ZF.CDU= 001000     |
| CB.SDB= 000010 | CS.MTL= 004000 | C.XPLI 000040  | FS.TRM= 003000    | ZF.DDM= 000001     |
| CB.SLI= 000100 | CS.RNG= 000010 | C.XPT 000034   | FS.WLB= 001000    | ZF.DIA= 004000     |
| CB.XLB= 000001 | CS.ROV= 000004 | C.XSVC 000042  | FS.XKL= 002000    | ZF.DLC= 000002     |
| CC.LLC= 000200 | CS.RSN= 010000 | C.XTC 000037   | FS.XOF= 010000    | ZF.DVP= 100000     |
| CE.ABO= 100362 | CS.SHU= 000001 | C.X25 000036   | FS.XON= 007000    | ZF.INI= 040000     |
| CE.DAO= 100346 | CS.SID= 000002 | DSSBUG= 177514 | FS.ZER= 002000    | ZF.KMX= 000020     |
| CE.DIS= 100366 | CS.STR= 000004 | DSSISK= 000000 | FSSLVL= 000001    | ZF.LLC= 000004     |
| CE.ERR= 100370 | CS.SUC= 000001 | DSSL11= 000001 | GSSTPP= 000000    | ZF.LMC= 000100     |
| CE.ILN= 100350 | CS.TMO= 020000 | DSSYNC= 000000 | GSSTSS= 000000    | ZF.MAN= 020000     |
| CE.LTO= 100356 | CS.XUR= 000004 | DSSYNM= 000000 | GSSTTK= 000000    | ZF.MFL= 000010     |
| CE.MOP= 100372 | CSSCKP= 000000 | E\$XPR= 000000 | GSSWRD= 000000    | ZF.MTM= 000400     |
| CE.NTE= 100361 | CSSORE= 000400 | FC.CCP= 000020 | I\$SRR= 000000    | ZF.MUX= 000040     |
| CE.RTE= 100376 | CSSRSH= 177564 | FC.CTL= 000006 | I\$SRDN= 000000   | ZF.PSE= 002000     |
| CE.SRC= 100364 | C.ADD 000034   | FC.KCP= 000016 | KISAR5= *****     | ZF.SLI= 010000     |
| CE.STP= 100352 | C.AR5 000014   | FC.KIL= 000004 | KISAR6= *****     | ZF.TIM= 000200     |
| CE.TME= 100354 | C.AST 000012   | FC.MAN= 000024 | KSSCNT= 177546    | ZF.X3P= 000000     |
| CE.TMO= 100374 | C.BID 000003   | FC.MLD= 000026 | KSSCSR= 177546    | ZS.ASN= 100000     |
| CE.UNS= 100344 | C.BUF 000014   | FC.PCT= 000030 | KSSLDC= 000000    | ZS.BSY= 140000     |
| CF.CHN= 000001 | C.BUF1 000014  | FC.PWR= 000022 | KSTPS= 000074     | Z.AVL 000014       |
| CF.EOM= 000004 | C.BUF2 000024  | FC.RCE= 000002 | LD\$LP = 000000   | Z.DAT 000016       |
| CF.HDR= 000020 | C.CNT 000020   | FC.RCP= 000014 | L\$ASG= 000000    | Z.DSP 000000       |
| CF.LB = 100000 | C.CNT1 000020  | FC.TIM= 000010 | L\$DRV= 000000    | Z.FLG 000010       |
| CF.LIN= 000002 | C.CNT2 000030  | FC.XCP= 000012 | L\$P11= 000001    | Z.LEN = 000016     |
| CF.SOM= 000010 | C.CSTP= 000012 | FC.XME= 000000 | L\$P11R= 000000   | Z.LLN 000006       |
| CF.SYN= 000040 | C.DST 000016   | FS.AST= 000000 | M\$S\$CRB= 000124 | Z.MAP 000020       |
| CF.TRN= 000100 | C.EFN 000003   | FS.CIB= 002000 | M\$S\$CRX= 000000 | Z.NAM 000004       |
| CM.CIR= 000002 | C.FLG 000022   | FS.CRA= 001000 | M\$S\$FCS= 000000 | Z.PCB 000012       |
| CM.FMT= 100000 | C.FLG1 000022  | FS.DIS= 013000 | M\$S\$MGE= 000000 | Z.SCH 000007       |
| CM.HRD= 000002 | C.FLG2 000032  | FS.DVC= 001000 | M\$S\$NET= 000000 | \$CALLX 000272RG   |
| CM.LIN= 000000 | C.FNC 000010   | FS.ENB= 012000 | M\$S\$OVR= 000000 | \$CEACC 000400RG   |
| CM.LOO= 000001 | C.LGTH= 000020 | FS.EXI= 001000 | N\$S\$ACC= 000001 | \$CECAC 000406RG   |
| CM.XLO= 000004 | C.LIN 000006   | FS.GET= 006000 | N\$S\$BUF= 000001 | \$CEDIV 000542RG   |
| CP.DCF= 000040 | C.LNK 000000   | FS.HLT= 000000 | N\$S\$LDV= 000001 | \$CEMUL 000536RG   |
| CP.HDI= 000007 | C.MOD 000011   | FS.INI= 000000 | N\$S\$MCP= 000001 | \$CMPDVB= ***** GX |
| CP.FS = 177400 | C.MRKT= 000000 | FS.KIL= 000000 | N\$S\$MLL= 000001 | \$CMGIN 000000RG   |
| CP.PSI= 000200 | C.NSP 000004   | FS.LCL= 100000 | N\$S\$MOV= 000010 | \$CMGRM 000044RG   |
| CP.XCF= 000100 | C.PRO 000042   | FS.LTM= 001000 | N\$S\$NCT= 000001 | \$CNV18 000114RG   |
| CP.ZFR= 000030 | C.RQT 000002   | FS.MNT= 004000 | N\$S\$PEM= 000001 | \$CNV22 000114RG   |
| CS.ABO= 000100 | C.RSI 000012   | FS.MSN= 014000 | OFS = 000006      | \$SMVFB 000216RG   |
| CS.BRO= 000002 | C.RSV 000002   | FS.REA= 001000 | P\$P45= 000000    | \$SMVBF 000152RG   |
| CS.BUF= 000200 | C.SCHD= 000002 | FS.RET= 000000 | P\$S\$WRD= 000000 | \$SPVID 000460RG   |
| CS.CES= 000002 | C.SRC 000014   | FS.REZ= 003000 | Q\$SOPT= 000010   | \$SPVNM= ***** GX  |
| CS.CHN= 000010 | C.SSHT= 000004 | FS.RLB= 002000 | R\$S\$DER= 000000 | \$SPVTA= ***** GX  |
| CS.CMP= 000200 | C.STA 000007   | FS.RNG= 011000 | R\$S\$EIS= 000001 | \$PUMR = ***** GX  |
| CS.DCR= 000400 | C.STS 000012   | FS.RST= 000000 | R\$S\$K11= 000001 | \$XBIAS= ***** GX  |
| CS.DEF= 000004 | C.SUB 000012   | FS.RTN= 001000 | R\$S\$ND= 000000  | .\$\$\$\$= 000034  |
| CS.DEV= 000002 | C.SYST= 000006 |                |                   |                    |

. ABS. 177776 000 (RW,1,GBL,ABS,DVR)

CETIM      CREATED BY MACRO ON 28-JUN-85 AT 18:21      PAGE 1      F 3  
 SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL | VALUE    | REFERENCES |
|--------|----------|------------|
| ISSAS  | = *****  | 4-58       |
| KISARS | = *****  | 5-82       |
| LF.ACT | = 100000 | #4-57      |
| LF.BRO | = 000400 | #4-57      |
| LF.BWT | = 000007 | #4-57      |
| LF.ENA | = 002000 | #4-57      |
| LF.LPB | = 001000 | #4-57      |
| LF.MDC | = 000100 | #4-57      |
| LF.MFL | = 004000 | #4-57      |
| LF.MTP | = 000020 | #4-57      |
| LF.PAC | = 000200 | #4-57      |
| LF.RDY | = 040000 | #4-57      |
| LF.REA | = 010000 | #4-57      |
| LF.SER | = 000040 | #4-57      |
| LF.TIM | = 000010 | #4-57      |
| LF.UNL | = 020000 | #4-57      |
| LF.X2P | = 000000 | #4-57      |
| LN.CLO | = 000000 | #4-57      |
| LN.DUM | = 000005 | #4-57      |
| LN.LOA | = 000004 | #4-57      |
| LN.LCO | = 000003 | #4-57      |
| LN.OAU | = 000003 | #4-57      |
| LN.OFF | = 000001 | #4-57      |
| LN.ON  | = 000000 | #4-57      |
| LN.OOP | = 000004 | #4-57      |
| LN.OPE | = 000001 | #4-57      |
| LN.REF | = 000002 | #4-57      |
| LN.SER | = 000002 | #4-57      |
| LN.STA | = 000017 | #4-57      |
| LN.SUB | = 000360 | #4-57      |
| LN.TRI | = 000006 | #4-57      |
| L.COST | = 000015 | #4-57      |
| L.CTL  | = 000012 | #4-57      |
| L.CVA  | = 177776 | #4-57      |
| L.DDM  | = 000002 | #4-57      |
| L.DDS  | = 000004 | #4-57      |
| L.DLC  | = 000003 | #4-57      |
| L.DLM  | = 000006 | #4-57      |
| L.DLS  | = 000010 | #4-57      |
| L.FLG  | = 000000 | #4-57      |
| L.KRBA | = 000016 | #4-57      |
| L.LEN  | = 000022 | #4-57      |
| L.MPF  | = 000022 | #4-57      |
| L.NMST | = 000020 | #4-57      |
| L.NSTA | = 000014 | #4-57      |
| L.OWNR | = 000021 | #4-57      |
| L.UNT  | = 000013 | #4-57      |
| M\$MGE | = 000000 | 5-95       |
| PR7    | = *****  | 5-87       |
| PS     | = *****  | 5-87       |
| R\$11D | = *****  | 4-58       |
| R\$11M | = 000000 | 4-58       |

CEXCM CREATED BY MACRU ON 28-JUN-85 AT 18:21

PAGE 1 F 4

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE      | REFERENCES |
|----------|------------|------------|
| CF.DDM   | = 000002   | #4-58      |
| CF.DYN   | = 000004   | #4-58      |
| CF.EIS   | = 000010   | #4-58      |
| CF.FRK   | = 100000   | #4-58      |
| CF.LOG   | = 000020   | #4-58      |
| CF.MDM   | = 000001   | #4-58      |
| CF.TIM   | = 000400   | #4-58      |
| CRCOV    | = 000002 G | #4-66      |
| DLCOV    | = 000010 G | #4-66      |
| FC.PWR   | = 000022   | 5-149      |
| FC.TIM   | = 000010   | 5-112      |
| FS.LTM   | = 001000   | 5-129      |
| FS.STM   | = 000000   | 5-112      |
| MSSNET   | = 000000   | 4-50       |
| MSSPRO   | = *****    | 5-94       |
| RSSMPL   | = *****    | 4-1        |
|          |            | 5-317      |
|          |            | 5-123      |
|          |            | 4-2        |
|          |            | 5-330      |
|          |            | 4-52       |
|          |            | 5-111      |
|          |            | 5-128      |
|          |            | 5-148      |
|          |            | 5-171      |
|          |            | 5-210      |
|          |            | 5-270      |
| \$CCBAF  | 000030 RG  | #5-86      |
| \$CCBAL  | 000034 RG  | #5-88      |
| \$CCBCT  | 000026 RG  | #5-85      |
| \$CCBLH  | 000134 RG  | #5-154     |
| \$CCBNM  | 000012 RG  | #5-79      |
| \$CCBSZ  | 000014 RG  | #5-80      |
| \$CEAVL  | 000002 RG  | #5-321     |
| \$CELFN  | 000304 RG  | #5-205     |
| \$CMFRK  | 000046 RG  | #5-98      |
| \$CMPDV  | 000040 RG  | #5-90      |
| \$CXOPT  | 000164 RG  | #5-166     |
| \$DDFNC  | 000132 RG  | #5-147     |
| \$DECPT  | 000124 RG  | #5-143     |
| \$EMASK  | 000314 RG  | #5-262     |
| \$ETIMR  | 000314 RG  | #5-263     |
| \$EVDSC  | 000226 RG  | #5-193     |
| \$FILHD  | 000222 RG  | #5-190     |
| \$IMASK  | 000314 RG  | #5-261     |
| \$LDBAF  | 000032 RG  | #5-87      |
| \$LGCOM  | 000300 RG  | #5-203     |
| \$LGDOB  | 000220 RG  | #5-189     |
| \$LGFINB | 000246 RG  | #5-195     |
| \$LGMON  | 000274 RG  | #5-202     |
| \$LGPDV  | 000216 RG  | #5-188     |
| \$LGSTT  | 000224 RG  | #5-191     |
| \$LGUIC  | 000272 RG  | #5-201     |
| \$LLCTA  | 000004 RG  | #5-76      |
| \$LTMFC  | 000104 RG  | #5-127     |
| \$MAXOV  | 000214 RG  | #5-186     |
| \$NBIAS  | 000166 RG  | #5-168     |
| \$NMCLH  | 000306 RG  | #5-207     |
| \$NMCL2  | 000310 RG  | #5-208     |
| \$NMLST  | 000204 RG  | #5-178     |
| \$NTLPT  | 000162 RG  | #5-165     |
| \$OBJHD  | 000206 RG  | #5-179     |



STCRC - MACRO V05.03b Friday 28-Jun-85 18:21 Page 9-1  
\$STCRC - CALCULATE CRC ON BLOCK OF DATA

F 5

272  
273  
274 000160

.ENDC ; M\$\$MGE  
RETURN

STCRC      CREATED BY    MACRO    ON 28-JUN-85 AT 18:22

PAGE 2    F 6

SYMBOL CROSS REFERENCE

CREF    04.00

SYMBOL    VALUE

REFERENCES

|         |            |        |       |
|---------|------------|--------|-------|
| \$STCVL | = 000002 G | #7-107 |       |
| \$STCVI | 000000 RG  | #7-103 | 7-107 |
| .BASEB  | = 140000   | #5-68  |       |

276  
 277  
 278  
 279  
 280  
 281  
 282  
 283  
 284  
 285  
 286  
 287  
 288  
 289  
 290  
 291 000162  
 292  
 293  
 294  
 295  
 296  
 297  
 298  
 299  
 300  
 301  
 302  
 303  
 304  
 305  
 306  
 307  
 308  
 309  
 310  
 311  
 312  
 313  
 314  
 315  
 316  
 317  
 318  
 319  
 320  
 321  
 322  
 323  
 324  
 325  
 326  
 327  
 328  
 329  
 330  
 331  
 332

```

.SBTTL STCR2 - CALCULATE CRC
;+
;**-STCR2-CALCULATE CRC FOR A BLOCK OF DATA
;
;CALCULATE THE CRC-16 FOR A BLOCK OF DATA USING THE KG-11 (IF AVAILABLE)
;OR SOFTWARE.
;-
;INPUTS:
;R0 - # OF BYTES IN DATA BLOCK
;R1 - INITIAL CRC VALUE
;R2 - POINTER TO DATA BLOCK
;
;OUTPUTS:
;R1 - UPDATED CRC VALUE
STCR2: SAVRG <R0,R2,R4,R5> ; SAVE SOME REGISTERS
;
;.IF DF K$$G11
;
;MOV #KGCSR,R4 ; POINT TO KG-11 CSR
;MOV #KGCSR+4,R5 ; POINT TO KG-11 DATA REGISTER
;MOV (R4)+,-(SP) ; SAVE CURRENT STATUS OF KG-11
;MOV (R4)+,-(SP) ; SINCE IT IS SHARED
;MOV #KGLDBC,-(R4) ; SET UP TO LOAD NEW BCC
;
10$: TSTB (R4) ; WAIT FOR COMPLETION
; BPL 10$; (ONLY ON FASTER PROCESSORS)
;
;MOV R1,(R5) ; LOAD NEW CRC
;MOV #KGINIT,(R4) ; AND INIT FOR CRC-16
;
20$: TSTB (R4) ; WAIT FOR COMPLETION
; BPL 20$; ...
;
;BIT #1,R2 ; DOES BUFFER START ON ODD BOUNDARY?
;BEQ 40$; IF EQ, NO
;BIC #DDB,(R4) ; SET KG-11 TO BYTE MODE
;MOVB (R2)+,(R5) ; AND DO CRC ON BYTE
;35$: TSTB (R4) ; Wait for completion
; BPL 35$; ...
;BIS #DDB,(R4) ; RESET WORD MODE
;R0 ; REDUCE BYTE COUNT
;BLE 80$; IF LE, ALL DONE
;
40$: ASR R0 ; CONVERT TO WORD COUNT
; ROR -(SP) ; SAVE ODD BYTE COUNT ON STACK
;
50$: DEC R0 ; ANY MORE WORDS TO GO?
; BLT 70$; IF LT, NO
;MOV (R2)+,(R5) ; DO CRC ON WORD
;
60$: TSTB (R4) ; WAIT FOR COMPLETION
; BMI 50$; IF MI, ALL DONE
;BR 60$; CONTINUE WAITING
;
70$: TST (SP)+ ; CHECK FOR ODD BYTE COUNT
; BMI 30$; IF MI, COMPUTE CRC ON LAST BYTE

```

STCRF      CREATED BY    MACRO    ON 28-JUN-85 AT 18:22

PAGE 3      F 8

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|         |       |        |        |
|---------|-------|--------|--------|
| CALL    | 9-185 | 10-260 |        |
| CCBDF\$ | #6-65 | 6-67   |        |
| NHWDF\$ | #6-65 | 6-68   |        |
| RESRG   | #6-65 | 9-199  | 11-428 |
| RETURN  | 9-213 | 10-274 | 11-429 |
| SAVRG   | #6-65 | 9-145  | 11-291 |

```

333 000304 016501 177776 80$: MOV -2(R5),R1 ; GET NEW CRC FROM KG-11
334 000310 012714 000133 MOV #KGLDBC,(R4) ; SET UP TO RELOAD OLD BCC
335
336 000314 105714 90$: TSTB (R4) ; WAIT FOR COMPLETION
337 000316 100376 BPL 90$; ...
338
339 000320 012615 MOV (SP)+,(R5) ; RESTORE OLD BCC
340 000322 012614 MOV (SP)+,(R4) ; AND OLD CSR STATUS
341
342 .IFF ; K$$G11
343
344 .IF NDF R$$E1S
345
346 SAVRG <R3> ; SAVE AN EXTRA REGISTER
347
348 .ENDC
349
350 5$:
351 .IF DF F$$AST
352
353 10$: MOVB (R2)+,R5 ; GET NEXT BYTE FROM DATA BLOCK
354
355 XOR R1,R5 ; EXCLUSIVE 'OR' BYTE WITH OLD CRC
356 BIC #^C<377>,R5 ; ISOLATE USEFUL BYTE
357 ASL R5 ; FORM WORD ADDRESS
358
359 .IF DF M$$MGE
360
361 MOV CTABL(R5),R5 ; GET MODIFIER WORD FROM TABLE
362
363 .IFF
364
365 ADD PC,R5 ; FORM PIC ADDRESS
366 MOV CTABL-,(R5),R5 ; GET MODIFIER WORD FROM TABLE
367
368 .ENDC
369
370 CLRB R1 ; CLEAR LOW BYTE OF OLD CRC
371 SWAB R1 ; NOW WORK ON UPPER BYTE
372 XOR R5,R1 ; PRODUCE UPDATED CRC
373
374 .IFF ; DF F$$AST
375
376 10$: MOVB (R2)+,R5 ; GET NEXT BYTE FROM DATA BLOCK
377
378 XOR R1,R5 ; EXCLUSIVE 'OR' BYTE WITH OLD CRC
379 MOV R5,R4 ; SAVE A COPY OF THE RESULT
380 BIC #^C<17>,R5 ; EXTRACT LOW 4 BITS
381 ASL R5 ; FORM WORD ADDRESS
382
383 .IF DF M$$MGE
384
385 MOV CTABL(R5),R5 ; GET FIRST MODIFIER WORD
386
387 .IFF
388
389 ADD PC,R5 ; FORM PIC ADDRESS

```

|                  |            |                  |     |     |     |
|------------------|------------|------------------|-----|-----|-----|
|                  | AAAAAAAAAA | UUU              | UUU | XXX | XXX |
|                  | AAAAAAAAAA | UUU              | UUU | XXX | XXX |
|                  | AAAAAAAAAA | UUU              | UUU | XXX | XXX |
| AAA              | AAA        | UUU              | UUU | XXX | XXX |
| AAA              | AAA        | UUU              | UUU | XXX | XXX |
| AAA              | AAA        | UUU              | UUU | XXX | XXX |
| AAA              | AAA        | UUU              | UUU | XXX | XXX |
| AAA              | AAA        | UUU              | UUU | XXX | XXX |
| AAA              | AAA        | UUU              | UUU | XXX | XXX |
| AAA              | AAA        | UUU              | UUU | XXX | XXX |
| AAA              | AAA        | UUU              | UUU | XXX | XXX |
| AAAAAAAAAAAAAAAA | UUU        | UUU              | XXX | XXX |     |
| AAAAAAAAAAAAAAAA | UUU        | UUU              | XXX | XXX |     |
| AAAAAAAAAAAAAAAA | UUU        | UUU              | XXX | XXX |     |
| AAA              | AAA        | UUU              | UUU | XXX | XXX |
| AAA              | AAA        | UUU              | UUU | XXX | XXX |
| AAA              | AAA        | UUU              | UUU | XXX | XXX |
| AAA              | AAA        | UUUUUUUUUUUUUUUU | XXX | XXX |     |
| AAA              | AAA        | UUUUUUUUUUUUUUUU | XXX | XXX |     |
| AAA              | AAA        | UUUUUUUUUUUUUUUU | XXX | XXX |     |

AXBFR      CREATED BY    MACRO    ON 28-JUN-85 AT 18:28      PAGE 4      F 11  
MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES                            |
|------------|---------------------------------------|
| CALL       | 6-156      6-170      6-178           |
| CCBDF\$    | #5-58      5-59                       |
| ENABL\$    | #5-58                                 |
| INHIS\$    | #5-58                                 |
| MTPS       | 6-96      6-106      6-166      6-186 |
| NHWDF\$    | #5-58      5-61                       |
| RETURN     | 6-200                                 |
| SLTDF\$    | #5-58      5-60                       |

```

58 : DECNET-11M/S V3.0
59 : DECNET-11M-PLUS V1.0
60 :
61 : 4.00 07-NOV-83
62 : DECNET-11M V4.0
63 : DECNET-11M-PLUS V2.0
64 :
65 : 5.00 22-JUL-85
66 : DECnet-11M/S V4.2
67 : DECnet-11M-Plus V3.0
68 : DECnet-Micro/RSX V1.0
69 :
70 : 5.01 09-Aug-85
71 : Add logic to bypass service disabled check on broadcast channels.
72 : This change requires EPMMAI ident V5.01.
73 :

```



AXDSP      CREATED BY    MACRO    ON 3-SEP-85 AT 10:58

PAGE 3    F 13

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|         |       |       |       |
|---------|-------|-------|-------|
| CALL    | 9-294 | 9-310 |       |
| CALLR   | 6-120 | 9-304 | 9-308 |
| CCBDF\$ | #5-75 | 5-78  |       |
| RESRG   | #5-75 | 9-311 |       |
| RETURN  | 6-119 | 9-313 |       |
| SAVRG   | #5-75 | 9-309 |       |
| SLTDF\$ | #5-75 | 5-77  |       |
| SOB     | 9-289 |       |       |

|                  |                  |                  |                 |                   |
|------------------|------------------|------------------|-----------------|-------------------|
| ASCMP = ***** GX | CS.DEV= 000002   | D\$SYNM= 000000  | I\$SRAR= 000000 | MDMCTL= ***** GX  |
| A\$CHK= 000000   | CS.DIS= 000040   | EXRON = ***** GX | I\$SRDN= 000000 | MDMSCN= ***** GX  |
| A\$CPS= 000000   | CS.ENA= 000001   | E\$XPR= 000000   | KMCL 000302R    | M\$CRB= 000124    |
| A\$PRI= 000000   | CS.ENB= 000020   | FC.CCP= 000020   | K\$CNT= 177546  | M\$CRX= 000000    |
| A\$TRP= 000000   | CS.ERR= 100000   | FC.CTL= 000006   | K\$CSR= 177546  | M\$SCS= 000000    |
| CB.CCB= 000002   | CS.FTL= 001000   | FC.KCP= 000016   | K\$LDL= 000000  | M\$MGE= 000000    |
| CB.DDM= 000040   | CS.HCR= 000001   | FC.KIL= 000004   | K\$TPS= 000074  | M\$NET= 000000    |
| CB.DLC= 000020   | CS.HFE= 002000   | FC.MAN= 000024   | LD\$LP= 000000  | M\$OVR= 000000    |
| CB.RDB= 000004   | CS.LST= 040000   | FC.MLD= 000026   | LF.ACT= 100000  | NETACP 000306R    |
| CB.SDB= 000010   | CS.MTL= 004000   | FC.PCT= 000030   | LF.BRO= 000400  | NMCL2 = ***** GX  |
| CB.SLI= 000100   | CS.RNG= 000010   | FC.PWR= 000022   | LF.BWT= 000007  | NMCMPL 000312R    |
| CB.XLB= 000001   | CS.ROV= 000004   | FC.RCE= 000002   | LF.ENA= 002000  | N\$SACC= 000001   |
| CCBRET 000030R   | CS.RSN= 010000   | FC.RCP= 000014   | LF.LPB= 001000  | N\$SBUF= 000001   |
| CCBRT = ***** GX | CS.SHU= 000001   | FC.TIM= 000010   | LF.MDC= 000100  | N\$SLDV= 000001   |
| CC.LLC= 000200   | CS.SID= 000002   | FC.XCP= 000012   | LF.MFL= 004000  | N\$SMCP= 000001   |
| CE.ABO= 100362   | CS.STR= 000004   | FC.XME= 000000   | LF.MTP= 000020  | N\$SMML= 000001   |
| CE.DAO= 100346   | CS.SUC= 000001   | FS.AST= 000000   | LF.PAC= 000200  | N\$SMOV= 000010   |
| CE.DIS= 100366   | CS.TMO= 020000   | FS.CIB= 002000   | LF.RDY= 040000  | N\$SNCT= 000001   |
| CE.ERR= 100370   | CS.XUR= 000004   | FS.CRA= 001000   | LF.REA= 010000  | N\$SPEM= 000001   |
| CE.ILN= 100350   | C\$CKP= 000000   | FS.DIS= 013000   | LF.SER= 000040  | PDSPIL = ***** GX |
| CE.LTO= 100356   | C\$ORE= 000400   | FS.DVC= 001000   | LF.TIM= 000010  | PWFAIL 000034R    |
| CE.MPO= 100372   | C\$RSH= 177564   | FS.ENB= 012000   | LF.UNL= 020000  | PWRFL 000172R     |
| CE.NTE= 100361   | C.ADD 000034     | FS.EXI= 001000   | LF.X2P= 000000  | PWRF1 = ***** GX  |
| CE.RTE= 100376   | C.BID 000003     | FS.GET= 006000   | LN.CLO= 000000  | P\$P45= 000000    |
| CE.SRC= 100364   | C.BUF 000014     | FS.HLT= 000000   | LN.DUM= 000005  | P\$RFL= 000001    |
| CE.STP= 100352   | C.BUF1 000014    | FS.INI= 000000   | LN.LOA= 000004  | P\$WRD= 000000    |
| CE.TMC= 100354   | C.BUF2 000024    | FS.KIL= 000000   | LN.LQJ= 000003  | Q\$OPT= 000010    |
| CE.TMO= 100374   | C.CNT 000020     | FS.LCL= 100000   | LN.OAU= 000003  | RDBRT = ***** GX  |
| CE.UNS= 100344   | C.CNT1 000020    | FS.LTM= 001000   | LN.OFF= 000001  | R\$DER= 000000    |
| CF.CHN= 000001   | C.CNT2 000030    | FS.MNT= 004000   | LN.ON= 000000   | R\$K11= 000001    |
| CF.EOM= 000004   | C.FLG 000022     | FS.MSN= 014000   | LN.OOP= 000004  | R\$SND= 000000    |
| CF.HDR= 000020   | C.FLG1 000022    | FS.REA= 001000   | LN.OPE= 000001  | R\$S11M= 000000   |
| CF.LB= 100000    | C.FLG2 000032    | FS.RET= 000000   | LN.REF= 000002  | SF.ACT= 000200    |
| CF.LIN= 000002   | C.FNC 000010     | FS.REZ= 003000   | LN.SER= 000002  | SF.ENA= 000100    |
| CF.SOM= 000010   | C.LIN 000006     | FS.RLB= 002000   | LN.STA= 000017  | SF.LPB= 000004    |
| CF.SYN= 000040   | C.LNK 000000     | FS.RNG= 011000   | LN.SUB= 000360  | SF.MFL= 000040    |
| CF.TRN= 000100   | C.MOD 000011     | FS.RST= 000000   | LN.TRI= 000006  | SF.PAC= 000020    |
| CM.CIR= 000002   | C.NSP 000004     | FS.RTN= 001000   | L\$ASG= 000000  | SF.REA= 000010    |
| CM.FMT= 100000   | C.PRO 000042     | FS.SET= 005000   | L\$DRV= 000000  | SF.SER= 000001    |
| CM.HRD= 000002   | C.RSV 000002     | FS.SF= 005000    | L\$P11= 000001  | SF.SVC= 000002    |
| CM.LIN= 000000   | C.STA 000007     | FS.SFR= 006000   | L\$S11R= 000000 | SF.UNL= 000040    |
| CM.LOO= 000001   | C.STS 000012     | FS.SFS= 004000   | L.COST 000015   | SLTMA = ***** GX  |
| CM.XLO= 000004   | C.URM 177776     | FS.SPW= 040000   | L.CTL 000012    | SLTNM = ***** GX  |
| CP.DCF= 000040   | C.XACP 000004    | FS.STM= 000000   | L.CVA 177776    | SRSTD = ***** GX  |
| CP.HDL= 000007   | C.XID 000035     | FS.STP= 002000   | L.DDM 000002    | S\$WRG= 000000    |
| CP.PS= 177400    | C.XLEN 000044    | FS.STR= 001000   | L.DDS 000004    | S\$YSZ= 007600    |
| CP.PSI= 000200   | C.XPLI 000040    | FS.TRM= 003000   | L.DLC 000003    | S.COST 000001     |
| CP.XCF= 000100   | C.XPT 000034     | FS.WLB= 001000   | L.DLM 000006    | S.FLG 000000      |
| CP.2FR= 000030   | C.XSVC 000042    | FS.XKL= 002000   | L.DLS 000010    | S.LEN 000004      |
| CS.ABO= 000100   | C.XTC 000037     | FS.XOF= 010000   | L.FLG 000000    | S.NMST 000002     |
| CS.BRO= 000002   | C.X25 000036     | FS.XON= 007000   | L.KRBA 000016   | S.OWNR 000003     |
| CS.BUF= 000200   | DDFNC = ***** GX | FS.ZER= 002000   | L.LEN= 000022   | S.OWNR 000003     |
| CS.CES= 000002   | DUMMY 000026R    | F\$SLVL= 000001  | L.MPF 000022    | TRIB 000170R      |
| CS.CHN= 000010   | D\$BUG= 177514   | G\$TPP= 000000   | L.NMST 000020   | TSKRT = ***** GX  |
| CS.CMP= 000200   | D\$ISK= 000000   | G\$TSS= 000000   | L.NSTA 000014   | T\$KMG= 000000    |
| CS.DCR= 000400   | D\$SLT1= 000001  | G\$TTK= 000000   | L.OWNR 000021   | T\$MIN= 000000    |
| CS.DEF= 000004   | D\$SYNC= 000000  | G\$WRD= 000000   | L.UNT 000013    | V\$CTR= 001000    |
|                  |                  |                  |                 | X\$DBT= 000000    |

.SBTTL POWERFAIL RECOVERY DISPATCH TO DDM MODULES

\*\*\*PWRF1-POWERFAIL RECOVERY DISPATCH TO DDM MODULES

THIS ROUTINE IS INVOKED FROM THE DRIVER POWERFAIL ENTRY POINTS TO DISPATCH TO DDM MODULES TO PERFORM POWERFAIL RECOVERY. IF THE KMC MICROCODE LOADER TASK IS INSTALLED, IT WILL BE REQUESTED TO RUN.

```

193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249

PWRF1: MOV #1,@PWRF1 ; STOP TIMERS RUNNING TO DDM PROCESSES
 MOV @SLTNM,-(SP) ; SET UP COUNT OF LINES TO SCAN

10$: MOV (SP),R3 ; GET NEXT SYSTEM LINE #
 DEC R3 ;
 ASL R3 ; FORM WORD OFFSET
 ADD @SLTMA,R3 ; POINT INTO SYSTEM LINE INDEX TABLE
 MOV (R3),R3 ; AND GET ADDRESS OF SYSTEM LINE TABLE
 TST (R3) ; IS THE LINE ACTIVE?
 BPL 20$; IF PL, NO

 .IF DF R$$MPL
 .IF NDF R$$PRO

 BIT #F2.MP,@FMSK2 ; IS THIS A MULTIPROCESSOR?
 BEQ 15$; BR IF NO
 MOV L.KRBA(R3),R5 ; GET ADDRESS OF KRB
 MOV @CPURM,-(SP) ; GET ADDRESS OF CPU URM TABLE
 BIT K.URM(R5),@C(SP)+ ; ARE WE RUNNING ON THE CORRECT PROCESSOR?
 BEQ 20$; IF EQ, NO

15$: .ENDC
 .ENDC

 MOV L.DDS(R3),R5 ; GET LINE TABLE ADDRESS
 MOVB L.DDM(R3),R2 ; AND DDM PDV INDEX
 MOV DDFNC,R3 ; POINT TO POWERFAIL FUNCTION CODE
 CALL @PDSPL ; DISPATCH TO DDM

20$: DEC (SP) ; REDUCE COUNT OF LINES TO SCAN
 BNE 10$; LOOP IF MORE TO GO
 TST (SP)+ ; CLEAN UP THE STACK

 .IF NDF I$$AS

 MOV #KMCL,R3 ; POINT TO KMC LOADER TASK NAME
 CALL @SRSTD ; SCAN TO FIND TASK'S TCB
 BCS 30$; IF CS, TASK NOT INSTALLED
 CLR R1 ; CLEAR DEFAULT UIC
 CALL @TSKRT ; REQUEST TASK TO RUN

 .ENDC

30$: MOV @SLTNM,@PWRF1 ; START DISCONNECT NOTIFICATION ON ALL LINES
 RETUR

 .IF NDF I$$AS
KMCL: .RAD50 /KMCL../ ; TASK NAME OF KMC MICROCODE LOADER

```

75

76

77 000000

78 000000

.MCALL SLTDF\$,CCBDF\$,SAVRG,RESRG

SLTDF\$

CCBDF\$

; DEFINE SYSTEM LINE TABLE OFFSETS

; DEFINE CCB OFFSETS

```

101 .SBTTL $CMQRM - REMOVE A CHAIN OF CCBS FROM A LIST
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125 000044 000261 $CMQRM::SEC
126 000046 011304 MOV (R3),R4 ; SET C-BIT JUST IN CASE QUEUE IS EMPTY
127 000050 001420 BEQ 40$; GET ADDRESS OF FIRST CCB IN QUEUE
128 000052 SAVRG <R5,R4> ; RETURN IF QUEUE IS EMPTY
129 000056 010405 10$: MOV R4,R5 ; SAVE A REGISTER
130 000060 032765 040000 000012 BIT #CS.LST,C.STS(R5) ; COPY FIRST CCB IN CHAIN ADDRESS
131 000066 001002 BNE 20$; IS THIS THE LAST CCB IN THE CHAIN?
132 000070 011504 MOV (R5),R4 ; YES - UPDATE THE LISTHEAD
133 000072 001371 BNE 10$; NO - GET NEXT CCB - IS THIS THE LAST?
134
135 000074 011513 20$: MOV (R5),(R3) ; SET ADDRESS OF NEW FIRST CCB IN QUEUE
136 000076 001002 BNE 30$; BRANCH IF QUEUE IS NOT EMPTY
137 000100 010363 000002 MOV R3,2(R3) ; OTHERWISE CLOSE UP LIST
138 000104 005015 30$: CLR (R5) ; CLEAR LINK POINTER IN THE LAST CCB OF CHAIN
139 000106 RESRG <R4,R5> ; RESTORE REGISTER
140 000112 40$: RETURN ; RETURN

```

CESUB1 MACRO V05.03b Friday 28-Jun-85 18:20 Page 17-2  
Symbol table

000554 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 15909 Words ( 63 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:17.11  
SY: CESUB1.V2,[130,134]CESUB1/CR/~SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[130,10]V2,CESUB

CETIM CREATED BY MACRO ON 28-JUN-85 AT 18:21

PAGE 2 G 3

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE       | REFERENCES |
|---------|-------------|------------|
| R\$11S  | = *****     | 4-58       |
| SF.ACT  | = 000200    | #4-57      |
| SF.ENA  | = 000100    | #4-57      |
| SF.LPB  | = 000004    | #4-57      |
| SF.MFL  | = 000040    | #4-57      |
| SF.PAC  | = 000020    | #4-57      |
| SF.REA  | = 000010    | #4-57      |
| SF.SER  | = 000001    | #4-57      |
| SF.SVC  | = 000002    | #4-57      |
| SF.UNL  | = 000040    | #4-57      |
| S.COST  | = 000001    | #4-57      |
| S.FLG   | = 000000    | #4-57      |
| S.LEN   | = 000004    | #4-57      |
| S.NMST  | = 000002    | #4-57      |
| S.OWNR  | = 000003    | #4-57      |
| X\$MBCB | = *****     | 4-58       |
| ZF.COU  | = 001000    | #4-58      |
| ZF.DDM  | = 000001    | #4-58      |
| ZF.L.A  | = 004000    | #4-58      |
| ZF.DLC  | = 000002    | #4-58      |
| ZF.DVP  | = 100000    | #4-58      |
| ZF.INI  | = 040000    | #4-58      |
| ZF.KMX  | = 000020    | #4-58      |
| ZF.LLC  | = 000004    | #4-58      |
| ZF.LMC  | = 000100    | #4-58      |
| ZF.MAN  | = 020000    | #4-58      |
| ZF.MFL  | = 000010    | #4-58      |
| ZF.MTM  | = 000400    | #4-58      |
| ZF.MUX  | = 000040    | #4-58      |
| ZF.PSE  | = 002000    | #4-58      |
| ZF.SLI  | = 010000    | #4-58      |
| ZF.TIM  | = 000200    | #4-58      |
| ZF.X3P  | = 000000    | #4-58      |
| ZS.ASN  | = 100000    | #4-58      |
| ZS.BSY  | = 140000    | #4-58      |
| Z.AVL   | = 000014    | #4-58      |
| Z.DAT   | = 000016    | #4-58      |
| Z.DSP   | = 000000    | #4-58      |
| Z.FLG   | = 000010    | #4-58      |
| Z.LEN   | = 000016    | #4-58      |
| Z.LLN   | = 000006    | #4-58      |
| Z.MAP   | = 000020    | #4-58      |
| Z.NAM   | = 000004    | #4-58      |
| Z.PCB   | = 000012    | #4-58      |
| Z.SCH   | = 000007    | #4-58      |
| \$DPTM  | = 000052 RG | 5-92       |
| \$LTMFC | = ***** GX  | 6-119      |
| \$PDSPL | = ***** GX  | 6-120      |
| \$PDVTA | = ***** GX  | 5-81       |
| \$TSTJM | = 000000 RG | #5-78      |

4-58

4-58

#6-118

CEXCM CREATED BY MACRO ON 28-JUN-85 AT 18:21

PAGE 2 G 4

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                                      |
|---------|------------|-------------------------------------------------|
| \$PAD   | = 000377 G | #4-66 5-308 5-308 5-308 5-308 5-308 5-308 5-308 |
| \$PADB  | 000332 RG  | #5-300                                          |
| \$PADBF | 000340 RG  | 5-302                                           |
| \$PADKL | = 000012 G | #4-66 5-303 5-306                               |
| \$PADSH | = 000002 G | #4-66                                           |
| \$PAVL  | 000202 RG  | #5-176                                          |
| \$PBIAS | 000320 RG  | #5-266                                          |
| \$PDVNM | 000006 RG  | #5-77                                           |
| \$PDVTA | 000000 RG  | #5-74                                           |
| \$PSIPT | 000126 RG  | #5-144                                          |
| \$PUMR  | 000174 RG  | #5-172                                          |
| \$PWRF1 | 000176 RG  | #5-174                                          |
| \$QBIAS | 000170 RG  | #5-169                                          |
| \$QSTRT | 000172 RG  | #5-170                                          |
| \$RDBAF | 000154 RG  | #5-162                                          |
| \$RDBCT | 000146 RG  | #5-159                                          |
| \$RDBLH | 000150 RG  | #5-160                                          |
| \$RDBNM | 000016 RG  | #5-81                                           |
| \$RDBSZ | 000020 RG  | #5-82                                           |
| \$RDBTH | 000036 RG  | #5-89                                           |
| \$RDQCT | 000160 RG  | #5-164                                          |
| \$RDQSL | 000156 RG  | #5-163                                          |
| \$SDBAF | 000144 RG  | #5-158                                          |
| \$SDBCT | 000136 RG  | #5-155                                          |
| \$SDBLH | 000140 RG  | #5-156                                          |
| \$SDBNM | 000022 RG  | #5-83                                           |
| \$SDBSZ | 000034 RG  | #5-84                                           |
| \$SHLST | 000312 RG  | #5-259                                          |
| \$SLTMA | 000002 RG  | #5-75                                           |
| \$SLTNM | 000010 RG  | #5-78                                           |
| \$SNAPT | 000130 RG  | #5-145                                          |
| \$SPAR1 | 000314 RG  | #5-264                                          |
| \$SPAR2 | 000320 RG  | #5-267                                          |
| \$SORCM | 000212 RG  | #5-183                                          |
| \$STMFC | 000064 RG  | #5-110                                          |
| \$SYNB  | 000324 RG  | #5-295                                          |
| \$SYNBF | 000352 RG  | #5-297                                          |
| \$SYNC  | = 000226 G | #5-309 5-312 5-312 5-312 5-312 5-312 5-312      |
| \$SYNCT | = 000010 G | #4-66 5-298 5-310                               |
| \$TK50  | 000210 RG  | #5-182                                          |
| \$TK100 | 000210 RG  | #5-181                                          |
| \$T1SCL | 000106 RG  | #5-134                                          |
| \$T100C | 000066 RG  | #5-118                                          |
| \$T100Q | 000060 RG  | #5-107                                          |
| \$T50CL | 000066 RG  | #5-117                                          |
| \$T50Q  | 000060 RG  | #5-106                                          |
| \$XAVL  | 000202 RG  | #5-177                                          |
| \$XBIAS | 000200 RG  | #5-175                                          |
| \$ZTIME | 000042 RG  | #5-91                                           |
| \$ZTIM2 | 000044 RG  | #5-92                                           |



276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332

000162

```
.SBTTL STCR2 - CALCULATE CRC
+
**--STCR2-CALCULATE CRC FOR A BLOCK OF DATA
 CALCULATE THE CRC-16 FOR A BLOCK OF DATA USING THE KG-11 (IF AVAILABLE)
 OR SOFTWARE.
-
INPUTS:
 R0 - # OF BYTES IN DATA BLOCK
 R1 - INITIAL CRC VALUE
 R2 - POINTER TO DATA BLOCK
OUTPUTS:
 R1 - UPDATED CRC VALUE
STCR2: SAVRG <R0,R2,R4,R5> ; SAVE SOME REGISTERS
 .IF DF K$$G11
 MOV #KGCSR,R4 ; POINT TO KG-11 CSR
 MOV #KGCSR+4,R5 ; POINT TO KG-11 DATA REGISTER
 MOV (R4)+,-(SP) ; SAVE CURRENT STATUS OF KG-11
 MOV (R4),-(SP) ; SINCE IT IS SHARED
 MOV #KGLDBC,-(R4) ; SET UP TO LOAD NEW BCC
 10$: TSTB (R4) ; WAIT FOR COMPLETION
 BPL 10$; (ONLY ON FASTER PROCESSORS)
 MOV R1,(R5) ; LOAD NEW CRC
 MOV #KGINIT,(R4) ; AND INIT FOR CRC-16
 20$: TSTB (R4) ; WAIT FOR COMPLETION
 BPL 20$; ...
 BIT #1,R2 ; DOES BUFFER START ON ODD BOUNDARY?
 BEQ 40$; IF EQ, NO
 BIC #DDB,(R4) ; SET KG-11 TO BYTE MODE
 MOV (R2)+,(R5) ; AND DO CRC ON BYTE
 35$: TSTB (R4) ; Wait for completion
 BPL 35$; ...
 BIS #DDB,(R4) ; RESET WORD MODE
 DEC R0 ; REDUCE BYTE COUNT
 BLE 80$; IF LE, ALL DONE
 40$: ASR R0 ; CONVERT TO WORD COUNT
 ROR -(SP) ; SAVE ODD BYTE COUNT ON STACK
 50$: DEC R0 ; ANY MORE WORDS TO GO?
 BLT 70$; IF LT, NO
 MOV (R2)+,(R5) ; DO CRC ON WORD
 60$: TSTB (R4) ; WAIT FOR COMPLETION
 BMI 50$; IF MI, ALL DONE
 BR 60$; CONTINUE WAITING
 70$: TST (SP)+ ; CHECK FOR ODD BYTE COUNT
 BMI 30$; IF MI, COMPUTE CRC ON LAST BYTE
```

STCRC      CREATED BY    MACRO    ON 28-JUN-85 AT 18:22

PAGE 3      G 6

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|         |       |       |        |
|---------|-------|-------|--------|
| CALL    | 8-185 | 9-260 |        |
| CCBDF\$ | #5-65 | 5-67  |        |
| NHWDF\$ | #5-65 | 5-68  |        |
| RESRG   | #5-65 | 8-199 | 10-428 |
| RETURN  | 8-213 | 9-274 | 10-429 |
| SAVRG   | #5-65 | 8-145 | 10-291 |

```

333 80$: MOV -2(R5),R1 ; GET NEW CRC FROM KG-11
334 MOV #KGLDBC,(R4) ; SET UP TO RELOAD OLD BCC
335
336 90$: TSTB (R4) ; WAIT FOR COMPLETION
337 BPL 90$; ...
338
339 MOV (SP)+,(R5) ; RESTORE OLD BCC
340 MOV (SP)+,(R4) ; AND OLD CSR STATUS
341
342 .IFF ; K$$G11
343
344 .IF NDF R$$EIS
345
346 SAVRG <R3> ; SAVE AN EXTRA REGISTER
347
348 .ENDC
349
350 000172 5$:
351 .IF DF F$$AST
352
353 000172 112205 10$: MOVB (R2)+,R5 ; GET NEXT BYTE FROM DATA BLOCK
354
355 000174 074105 XOR R1,R5 ; EXCLUSIVE 'OR' BYTE WITH OLD CRC
356 000176 042705 177400 BIC #^C<377>,R5 ; ISOLATE USEFUL BYTE
357 000202 006305 ASL R5 ; FORM WORD ADDRESS
358
359 .IF DF M$$MGE
360
361 000204 016505 000234' MOV CTABL(R5),R5 ; GET MODIFIER WORD FROM TABLE
362
363 .IFF
364
365 ADD PC,R5 ; FORM PIC ADDRESS
366 MOV CTABL-.(R5),R5 ; GET MODIFIER WORD FROM TABLE
367
368 .ENDC
369
370 000210 105001 CLRB R1 ; CLEAR LOW BYTE OF OLD CRC
371 000212 000301 SWAB R1 ; NOW WORK ON UPPER BYTE
372 000214 074501 XOR R5,R1 ; PRODUCE UPDATED CRC
373
374 .IFF ; DF F$$AST
375
376 10$: MOVB (R2)+,R5 ; GET NEXT BYTE FROM DATA BLOCK
377
378 XOR R1,R5 ; EXCLUSIVE 'OR' BYTE WITH OLD CRC
379 MOV R5,R4 ; SAVE A COPY OF THE RESULT
380 BIC #^C<17>,R5 ; EXTRACT LOW 4 BITS
381 ASL R5 ; FORM WORD ADDRESS
382
383 .IF DF M$$MGE
384
385 MOV CTABL(R5),R5 ; GET FIRST MODIFIER WORD
386
387 .IFF
388
389 ADD PC,R5 ; FORM PIC ADDRESS

```

\*\*FILE\*\*ID\*\*STCRCK

```

SSSSSSSS TTTTTTTTTT CCCCCCCC RRRRRRRR CCCCCCCC KK KK
SSSSSSSS TTTTTTTTTT CCCCCCCC RRRRRRRR CCCCCCCC KK KK
SS TT CC RR RR CC KK KK
SS TT CC RR RR CC KK KK
SS TT CC RR RR CC KK KK
SSSSSS TT CC RRRRRRRR CC KKKKKK
SSSSSS TT CC RRRRRRRR CC KKKKKK
SS TT CC RR RR CC KK KK
SS TT CC RR RR CC KK KK
SS TT CC RR RR CC KK KK
SSSSSSSS TT CCCCCCCC RR RR CCCCCCCC KK KK
SSSSSSSS TT CCCCCCCC RR RR CCCCCCCC KK KK

```

```

....
....
....
....

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTT'TTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLL' LLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```

STCRCK MACRO V05.03b Friday 28-Jun-85 18:22 Page 11-2  
 STCR2 - CALCULATE CRC

```

390 MOV CTABL-.(R5),R5 ; GET FIRST MODIFIER WORD
391
392 .ENDC
393
394 BIC #*C<360>,R4 ; NOW EXTRACT HIGH 4 BITS
395 ASR R4 ; FORM WORD OFFSET
396 ASR R4 ;
397 ASR R4 ; ...
398
399 .IF DF M$$MGE
400
401 MOV CTABL+32.(R4),R4 ; GET SECOND MODIFIER WORD
402
403 .IFF
404
405 ADD PC,R4 ; FORM PIC ADDRESS
406 MOV CTABL+32-.(R4),R4 ; GET SECOND MODIFIER WORD
407
408 .ENDC
409
410 XOR R4,R5 ; EXCLUSIVE 'OR' MODIFIER WORDS
411 CLRB R1 ; CLEAR LOW BYTE OF OLD CRC
412 SWAB R1 ; NOW WORK ON UPPER BYTE
413 XOR R5,R1 ; PRODUCE UPDATED CRC
414
415 .ENDC
416
417 DEC R0 ; ANY MORE TO GO?
418 BGT 10$; IF GT, YES
419
420 .IF NDF R$$E1S
421
422 RESRG <R3>
423
424 .ENDC
425
426 .ENDC
427
428 RESRG <R5,R4,R2,R0>
429 RETURN

```

000324  
 000334

\*\*FILE\*\*ID\*\*AXBFR

```

AAAAAA XX XX BBBB8888 FFFFFFFF RRRRRRRR
AAAAAA XX XX 88888888 FFFFFFFF RRRRRRRR
AA AA XX XX BB BB FF RR RR
AA AA XX XX BB BB FF RR RR
AA AA XX XX BB BB FF RR RR
AA AA XX XX BB BB FF RR RR
AA AA XX XX BB BB FF RR RR
AA AA XX XX BB BB FF RR RR
AAAAAAAA XX XX BB BB FF RR RR
AAAAAAAA XX XX BB BB FF RR RR
AA AA XX XX BB BB FF RR RR
AA AA XX XX BB BB FF RR RR
AA AA XX XX BB888888 FF RR RR
AA AA XX XX 88888888 FF RR RR

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLL SSSSSSSS TT
LLLLLLLL SSSSSSSS TT

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLLLL SSSSSSSS TT
LLLLLLLLLLL SSSSSSSS TT

```

75  
76  
77 000000  
78 000000

.MCALL SLTDF\$,CCBDF\$,SAVRG,RESRG

SLTDF\$  
CCBDF\$

; DEFINE SYSTEM LINE TABLE OFFSETS  
; DEFINE CCB OFFSETS



\*\*FILE\*\*ID\*\*AXDSPB

```

AAAAAA XX XX DDDDDDD SSSSSSS PPPPPPP BBBB8888
AAAAAA XX XX DDDDDDD SSSSSSS PPPPPPP 88888888
AA AA XX XX DD DD SS PP PP 88 88
AA AA XX XX DD DD SS PP PP 88 88
AA AA XX XX DD DD SS PP PP 88 88
AA AA XX XX DD DD SS PP PP 88 88
AA AA XX XX DD DD SS PP PP 88 88
AA AA XX XX DD DD SS PP PP 88 88
AAAAA8888 XX XX DD DD SS PP PP 88 88
AAAAA8888 XX XX DD DD SS PP PP 88 88
AA AA XX XX DD DD SS PP PP 88 88
AA AA XX XX DD DD SS PP PP 88 88
AA AA XX XX DDDDDDD SSSSSSS PP 88888888
AA AA XX XX DDDDDDD SSSSSSS PP 88888888

```

```

....
....
....
....

```

```

LL SSSSSSS TTTTTTTTT
LL SSSSSSS TTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLL SSSSSSS TT
LLLLLLLLL SSSSSSS TT

```

AXDSPB - AUXILLIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 10:58 <sup>G 14</sup> Page 10-3  
Symbol table

X\$\$MDC= 000001      \$AUXTB 000000RG      \$BFRTN= \*\*\*\*\* GX      .\$\$\$\$.= 000034

. ABS. 177776      000 (RW,I,GBL,ABS,OVR)  
         000516      001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 16133 Words ( 64 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:20.34  
DB2:AXDSPB.T47,[131,134]AXDSPB/CR/-SP=DB2:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]T47,AXDSP

AXDSPM - AUXILLIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 10:59 <sup>6 15</sup> Page 9-1

250  
251  
252

.ENDC

.ENDC

```

80
81
82
83
84 000000
85 000000 000001'
86 000002 000000G
87 000004 000005'
88
89
90
91
92
93
94
95
96 000006 000007'
97
98 000010 000034'
99
100
101
102
103 000012 000013'
104 000014 000015'
105 000016 000017'
106 000020 000030'
107
108 000022 000170'
109
110
111
112 000024 000310'
113
114
115
116 000026
117
118
119 000026
120 000030

```

```

;
; AUXILIARY PROCESS DISPATCH TABLE
;
$AUXTB::
 .WORD .+1 ; TRANSMIT ENABLE (NOP)
 .WORD $BFRTN ; RECEIVE ENABLE
 .WORD .+1 ; KILLIO (NOP)
 .IF DF X$$MDC
 .WORD MDMCTL ; MODEM CONTROL CONTROL ROUTINE
 .IF DF P$$RFL
 .WORD PWFAIL ; PERFORM POWERFAIL RECOVERY
 .IFF
 .WORD MDMSCN ; TIMEOUT ENTERS SCAN ROUTINE
 .ENDC
 .IFF
 .WORD .+1 ; CONTROL ENABLE (NOP)
 .IF DF P$$RFL
 .WORD PWFAIL ; PERFORM POWERFAIL RECOVERY
 .IFF
 .WORD DUMMY ; TIMEOUT (NOP)
 .ENDC
 .ENDC
 .WORD .+1 ; TRANSMIT COMPLETE (NOP)
 .WORD .+1 ; RECEIVE COMPLETE (NOP)
 .WORD .+1 ; KILL COMPLETE (NOP)
 .WORD CCBRET ; CONTROL COMPLETE - RELEASE CCB
 .IF DF P$$RFL
 .WORD PWRFL ; POWERFAIL DISPATCH TO DDM MODULES
 .IFF
 .WORD DUMMY ; POWERFAIL DISPATCH TO DDM MODULES (NOP)
 .ENDC
 .WORD NMCMP ; NETWORK MANAGEMENT COUNTER COMPLETION

 .IF NDF X$$MDC & R$$11D & I$$AS
$MDCIN::
 .ENDC
DUMMY: RETURN
CCBRET: CALLR @CCBRT

```

```

142 .SBTTL $CNV18 - CONVERT TO 18-BIT UNIBUS ADDRESS
143
144 ;+
145 ;**-$CNV18-CONVERT TO 18-BIT UNIBUS ADDRESS
146 ;**-$CNV22-CONVERT TO 22-BIT QBUS-22 ADDRESS
147
148 ; THIS SUBROUTINE IS CALLED TO CONVERT AN ADDRESS DOUBLEWORD
149 ; TO AN 18-BIT UNIBUS VIRTUAL ADDRESS. NOTE THAT THE SUBROUTINE
150 ; TO CONVERT TO A QBUS-22 ADDRESS IS THE SAME (AT THE MOMENT).
151
152 ; INPUTS:
153
154 ; R2 - VIRTUAL ADDRESS
155 ; R3 - RELOCATION BIAS (PHYSICAL ADDRESS/100)
156
157 ; OUTPUTS:
158
159 ; R2 - LOW ORDER 16 BITS OF UNIBUS ADDRESS
160 ; R3 - BITS 16 & 17 OF UNIBUS ADDRESS IN BITS 0 & 1
161
162 ; -
163
164 .ENABL LSB
165 $CNV22:: ; Reference Label
166 .IF DF,K$$DAS ; Save virtual address
167 MOV R2,-(SP) ; Buffer from exec, or common pool?
168 TST R3 ; If NE, from common pool - use bias supplied
169 BNE 5$; Get virtual in required register
170 MOV R2,R3 ; Set up for left shift
171 CLR R2 ; Shift active page field from virtual
172 ASHC #3,R2 ; Make it a word index
173 ASL R2 ; Calculate D-space APR address
174 ADD #KDSARQ,R2 ; Get contents of APR
175 MOV (R2),R3 ; Join common code
176 BR 5$
177 .ENDC ; DF,K$$DAS
178
179 $CNV18:: ; Reference Label
180 TST R3 ; Buffer from exec, or comm pool?
181 BEQ 10$; If EQ, it's from exec
182 ; (in the low 16 or 20 K)
183 ADD $PUMR,R3 ; Exec phys addr = virtual addr = unibus addr)
184 MOV R2,-(SP) ; Add starting UMR bias to block address
185 ; Save virtual
186
187 5$: BIC #160000,(SP) ; Clear APR selector from virtual address
188 CLR R2 ; Initialize 18 bit value
189 .IF DF R$$EIS ; ALIGN AS AN 18 BIT QUANTITY
190 ASHC #6,R2
191 .IFF ; DF R$$EIS
192
193 .REPT 6
194 ASL R3 ; SHIFT LOW ORDER BITS
195 ROL R2 ; INTO HIGH ORDER 2 BITS FOR 18-BIT VALUE
196 .ENDR
197
198 .ENDC ; DF R$$EIS
199
200 ADD R3,(SP) ; ADD LOW 16 BITS TO SAVED OFFSET IN BLOCK

```

CESUB1      CREATED BY MACRO    ON 28-JUN-85 AT 18:21      PAGE 1      H 2  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE      | REFERENCES                                                                                    |
|---------|------------|-----------------------------------------------------------------------------------------------|
| CS.LST  | = 040000   | 6-92      6-95      7-130                                                                     |
| C.STS   | 000012     | *6-92      *6-95      7-130                                                                   |
| IS\$AS  | = *****    | 5-64                                                                                          |
| KISAR5  | = ***** GX | 9-245      *9-246      10-294      *10-295      *10-303      11-327      *11-341      *11-361 |
| KISAR6  | = ***** GX | *12-390      *12-411                                                                          |
| K\$SDAS | = *****    | 8-164      11-323      11-324      11-329      11-343      11-357                             |
| M\$SPRO | = *****    | 16-515                                                                                        |
| OFS     | = 000006   | #11-323      11-350      *11-351                                                              |
| R\$EIS  | = 000001   | #4-2      5-1      5-2      8-186      12-398      14-470      15-501                         |
| R\$E11D | = *****    | 5-64                                                                                          |
| R\$E11M | = 000000   | 5-64                                                                                          |
| R\$E11S | = *****    | 5-64                                                                                          |
| X\$MBCB | = *****    | 5-64      5-64                                                                                |
| ZF.COU  | = 001000   | #5-64                                                                                         |
| ZF.DDM  | = 000001   | #5-64                                                                                         |
| ZF.DIA  | = 004000   | #5-64                                                                                         |
| ZF.DLC  | = 000002   | #5-64                                                                                         |
| ZF.DVP  | = 100000   | #5-64                                                                                         |
| ZF.INI  | = 040000   | #5-64                                                                                         |
| ZF.KMX  | = 000020   | #5-64                                                                                         |
| ZF.LLC  | = 000004   | #5-64                                                                                         |
| ZF.LMC  | = 000100   | #5-64                                                                                         |
| ZF.MAN  | = 020000   | #5-64                                                                                         |
| ZF.MFL  | = 000010   | #5-64                                                                                         |
| ZF.MTM  | = 000400   | #5-64                                                                                         |
| ZF.MUX  | = 000040   | #5-64                                                                                         |
| ZF.PSE  | = 002000   | #5-64                                                                                         |
| ZF.SLI  | = 010000   | #5-64                                                                                         |
| ZF.TIM  | = 000200   | #5-64                                                                                         |
| ZF.X3P  | = 000000   | #5-64                                                                                         |
| ZS.ASN  | = 100000   | #5-64                                                                                         |
| ZS.BSY  | = 140000   | #5-64                                                                                         |
| Z.AVL   | 000014     | #5-64                                                                                         |
| Z.DAT   | 000016     | #5-64                                                                                         |
| Z.DSP   | 000000     | #5-64      5-64                                                                               |
| Z.FLG   | 000010     | #5-64                                                                                         |
| Z.LEN   | = 000016   | #5-64                                                                                         |
| Z.LLN   | 000006     | #5-64                                                                                         |
| Z.MAP   | 000020     | #5-64                                                                                         |
| Z.NAM   | 000004     | #5-64      11-338      13-435                                                                 |
| Z.PCB   | 000012     | #5-64                                                                                         |
| Z.SCH   | 000007     | #5-64                                                                                         |
| \$CALLX | 000272 RG  | #11-326                                                                                       |
| \$CEACC | 000400 RG  | #12-390                                                                                       |
| \$CECAC | 000406 RG  | #12-391                                                                                       |
| \$CEDIV | 000542 RG  | #15-501                                                                                       |
| \$CEMUL | 000536 RG  | #14-470                                                                                       |
| \$CMPDV | = ***** GX | 11-326      *11-349      *11-362                                                              |
| \$CMQIN | 000000 RG  | #6-90                                                                                         |
| \$CMORM | 000044 RG  | #7-125                                                                                        |
| \$CNV18 | 000114 RG  | #8-176                                                                                        |
| \$CNV22 | 000114 RG  | #8-163                                                                                        |

CETIM      CREATED BY    MACRO    ON 28-JUN-85 AT 18:21

PAGE 3    H 3

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|         |       |                 |
|---------|-------|-----------------|
| CALL    | 5-92  | 6-120           |
| CCBDF\$ | #4-55 | 4-56            |
| ENABL\$ | #4-54 |                 |
| INHIB\$ | #4-54 |                 |
| MTP\$   | 5-87  | 5-93      6-122 |
| PDVDF\$ | #4-55 | 4-58            |
| RESRG   | #4-54 |                 |
| RETURN  | 5-101 | 6-123           |
| SAVRG   | #4-54 |                 |
| SLTDF\$ | #4-55 | 4-57            |

CEXCM      CREATED BY    MACRO    ON 28-JUN-85 AT 18:21

PAGE 3    H 4

MACRO CROSS REFERENCE

CREF    04.00

| MACRO NAME | REFERENCES |
|------------|------------|
|------------|------------|

|         |                 |
|---------|-----------------|
| CCBDF\$ | #4-54      4-56 |
| NKRDF\$ | #4-54      4-57 |
| OPTDF\$ | #4-54      4-58 |
| SYNDF\$ | #4-64      4-66 |



```

333 80$: MOV -2(R5),R1 ; GET NEW CRC FROM KG-11
334 MOV #KGLDBC,(R4) ; SET UP TO RELOAD OLD BCC
335
336 90$: ISTB (R4) ; WAIT FOR COMPLETION
337 BPL 90$; ...
338
339 MOV (SP)+,(R5) ; RESTORE OLD BCC
340 MOV (SP)+,(R4) ; AND OLD CSR STATUS
341
342 .IFF ; K$$G11
343
344 .IF NDF R$$EIS
345
346 SAVRG <R3> ; SAVE AN EXTRA REGISTER
347
348 .ENDC
349
350 000172 5$:
351 .IF DF F$$AST
352
353 10$: MOVB (R2)+,R5 ; GET NEXT BYTE FROM DATA BLOCK
354
355 XOR R1,R5 ; EXCLUSIVE 'OR' BYTE WITH OLD CRC
356 BIC #^C<37>,R5 ; ISOLATE USEFUL BYTE
357 ASL R5 ; FORM WORD ADDRESS
358
359 .IF DF M$$MGE
360
361 MOV CTABL(R5),R5 ; GET MODIFIER WORD FROM TABLE
362
363 .IFF
364
365 ADD PC,R5 ; FORM PIC ADDRESS
366 MOV CTABL-(R5),R5 ; GET MODIFIER WORD FROM TABLE
367
368 .ENDC
369
370 CLRB R1 ; CLEAR LOW BYTE OF OLD CRC
371 SWAB R1 ; NOW WORK ON UPPER BYTE
372 XOR R5,R1 ; PRODUCE UPDATED CRC
373
374 .IFF ; DF F$$AST
375
376 000172 112205 10$: MOVB (R2)+,R5 ; GET NEXT BYTE FROM DATA BLOCK
377
378 XOR R1,R5 ; EXCLUSIVE 'OR' BYTE WITH OLD CRC
379 MOV R5,R4 ; SAVE A COPY OF THE RESULT
380 BIC #^C<17>,R5 ; EXTRACT LOW 4 BITS
381 ASL R5 ; FORM WORD ADDRESS
382
383 .IF DF M$$MGE
384
385 MOV CTABL(R5),R5 ; GET FIRST MODIFIER WORD
386
387 .IFF
388
389 ADD PC,R5 ; FORM PIC ADDRESS

```

```

SSSSSSSS TTTTTTTTTT CCCCCCCC RRRRRRRR CCCCCCCC FFFFFFFF
SSSSSSSS TTTTTTTTTT CCCCCCCC RRRRRRRR CCCCCCCC FFFFFFFF
SS TT CC RR RR CC FF
SS TT CC RR RR CC FF
SS TT CC RR RR CC FF
SS TT CC RR RR CC FF
 SSSSSS TT CC RRRRRRRR CC FFFFFFFF
 SSSSSS TT CC RRRRRRRR CC FFFFFFFF
 SS TT CC RR RR CC FF
 SS TT CC RR RR CC FF
 SS TT CC RR RR CC FF
 SS TT CC RR RR CC FF
SSSSSSSS TT CCCCCCCC RR RR CCCCCCCC FF
SSSSSSSS TT CCCCCCCC RR RR CCCCCCCC FF

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLLLL SSSSSSSS TT
LLLLLLLLLLL SSSSSSSS TT

```

STCRF MACRO V05.03b Friday 28-Jun-85 18:22 Page 11-2  
 STCR2 - CALCULATE CRC

H 7

```

390 MOV CTABL-.(R5),R5 ; GET FIRST MODIFIER WORD
391
392 .ENDC
393
394 BIC #^C<360>,R4 ; NOW EXTRACT HIGH 4 BITS
395 ASR R4 ; FORM WORD OFFSET
396 ASR R4 ; ...
397 ASR R4 ; ...
398
399 .IF DF M$$MGE
400
401 MOV CTABL+32.(R4),R4; GET SECOND MODIFIER WORD
402
403 .IFF
404
405 ADD PC,R4 ; FORM PIC ADDRESS
406 MOV CTABL+32-.(R4),R4 ; GET SECOND MODIFIER WORD
407
408 .ENDC
409
410 XOR R4,R5 ; EXCLUSIVE 'OR' MODIFIER WORDS
411 CLRB R1 ; CLEAR LOW BYTE OF OLD CRC
412 SWAB R1 ; NOW WORK ON UPPER BYTE
413 XOR R5,R1 ; PRODUCE UPDATED CRC
414
415 .ENDC
416
417 000216 005300 DEC R0 ; ANY MORE TO GO?
418 000220 003364 BGT 10$; IF GT, YES
419 000222
20$:
420 .IF NDF R$$EIS
421
422 RESRG <R3>
423
424 .ENDC
425
426 .ENDC
427
428 000222 RESRG <R5,R4,R2,R0>
429 000232 RETURN

```

STCRCK MACRO V05.03b Friday 28-Jun-85 18:22  
Table of contents

|     |     |                                                |
|-----|-----|------------------------------------------------|
| 6-  | 63  | MACRO DEFINITIONS                              |
| 7-  | 90  | DEFINE KG-11 REGISTERS AND BITS                |
| 8-  | 101 | EXECUTIVE VECTOR TABLE                         |
| 9-  | 109 | \$CLCRC - CALCULATE CRC-16 ON A TRANSMIT CHAIN |
| 10- | 215 | \$STCRC - CALCULATE CRC ON BLOCK OF DATA       |
| 11- | 276 | STCR2 - CALCULATE CRC                          |

431  
 432  
 433  
 434  
 435  
 436  
 437  
 438  
 439  
 440  
 441  
 442  
 443  
 444  
 445  
 446  
 447  
 448  
 449  
 450  
 451  
 452  
 453  
 454  
 455  
 456  
 457  
 458  
 459  
 460  
 461  
 462  
 463  
 464  
 465  
 466  
 467  
 468  
 469  
 470  
 471  
 472  
 473  
 474  
 475  
 476  
 477  
 478  
 479  
 480  
 481  
 482  
 483  
 484  
 485  
 486  
 487

```
.IF NDF K$$$G11
.SBTTL MODIFIER TABLE FOR SOFTWARE CRC

CTABL: .WORD 0 ;FIRST HALF OF TABLE
 .WORD 140301
 .WORD 140601
 .WORD 500
 .WORD 141401
 .WORD 1700
 .WORD 1200
 .WORD 141101
 .WORD 143001
 .WORD 3300
 .WORD 3600
 .WORD 143501
 .WORD 2400
 .WORD 142701
 .WORD 142201
 .WORD 2100

 .IF DF F$$$AST
 .WORD 146001
 .WORD 6300
 .WORD 6600
 .WORD 146501
 .WORD 7400
 .WORD 147701
 .WORD 147201
 .WORD 7100
 .WORD 5000
 .WORD 145301
 .WORD 145601
 .WORD 5500
 .WORD 144401
 .WORD 4700
 .WORD 4200
 .WORD 144101
 .WORD 154001
 .WORD 14300
 .WORD 14600
 .WORD 154501
 .WORD 15400
 .WORD 155701
 .WORD 155201
 .WORD 15100
 .WORD 17000
 .WORD 157301
 .WORD 157601
 .WORD 17500
 .WORD 156401
 .WORD 16700
 .WORD 16200
 .WORD 156101
 .WORD 12000
```

AXBFR MACRO V05.03b Friday 28-Jun-85 18:28  
Table of contents

H 10

5- 56 Macro definitions

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- 40

: COPYRIGHT (C) 1984, 1985 BY  
 : DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

```

; THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
; NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
; EQUIPMENT CORPORATION.

```

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF  
ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

## : MODULE DESCRIPTION

```

: AUX GLOBAL DATA AREA
:
: DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
: IDENT HISTORY:

```

```

1.00 15-OCT-84
 MODULE CREATION
5.00 22-JUL-85
 DECnet-11M/S V4.2
 DECnet-11M-Plus V3.0
 DECnet-Micro/RSX V1.0

```

```

80 ;
81 ; AUXILIARY PROCESS DISPATCH TABLE
82 ;
83 ;
84 000000 $AUXTB: .WORD .+1 ; TRANSMIT ENABLE (NOP)
85 000000 000001' .WORD $BFRTN ; RECEIVE ENABLE
86 000002 000000G .WORD .+1 ; KILLIO (NOP)
87 000004 000005' .IF DF X$$MDC ;
88 .WORD MDMCTL ; MODEM CONTROL CONTROL ROUTINE
89 .IF DF P$$RFL ;
90 .WORD PWFAIL ; PERFORM POWERFAIL RECOVERY
91 .IF P$RFL ;
92 .WORD MDMSCN ; TIMEOUT ENTERS SCAN ROUTINE
93 .ENDC
94 .IF
95 000006 000007' .WORD .+1 ; CONTROL ENABLE (NOP)
96 .IF DF P$$RFL ;
97 .WORD PWFAIL ; PERFORM POWERFAIL RECOVERY
98 .IF
99 000010 000026' .WORD DUMMY ; TIMEOUT (NOP)
100 .ENDC
101 .ENDC
102 .WORD .+1 ; TRANSMIT COMPLETE (NOP)
103 .WORD .+1 ; RECEIVE COMPLETE (NOP)
104 .WORD .+1 ; KILL COMPLETE (NOP)
105 .WORD CCBRET ; CONTROL COMPLETE - RELEASE CCB
106 .IF DF P$$RFL ;
107 .WORD PWRFL ; POWERFAIL DISPATCH TO DDM MODULES
108 .IF
109 000022 000026' .WORD DUMMY ; POWERFAIL DISPATCH TO DDM MODULES (NOP)
110 .ENDC
111 000024 000040' .WORD NMCMP ; NETWORK MANAGEMENT COUNTER COMPLETION
112 .IF NDF X$$MDC & R$$11D & I$$AS
113 $MDCIN: .ENDC
114
115 DUMMY: RETURN
116 CCBRET: CALLR @CCBRT
117
118 000026
119 000030
120

```



AXDSPB - AUXILLIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 10:58 H 13  
Table of contents

|     |     |                                            |
|-----|-----|--------------------------------------------|
| 8-  | 123 | POWERFAIL RECOVERY ROUTINE                 |
| 9-  | 193 | POWERFAIL RECOVERY DISPATCH TO DDM MODULES |
| 10- | 254 | NETWORK MANAGEMENT COUNTER COMPLETION      |

AXDSPB CREATED BY MACRO ON 3-SEP-85 AT 10:58

PAGE 1 H 14

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL | VALUE      | REFERENCES            |
|--------|------------|-----------------------|
| ASCOMP | = ***** GX | 8-168                 |
| CB.CCB | = 000092   | 10-306                |
| CCBRET | 000030 R   | 7-106 #7-120          |
| CCBRT  | = ***** GX | 7-120 10-308          |
| CE.DIS | = 100366   | 8-167                 |
| CS.LST | = 040000   | 10-303                |
| C.ADD  | 000034     | 10-270                |
| C.B.D  | 000003     | 10-306                |
| C.CNT2 | 000030     | 10-274 10-286         |
| C.STS  | 000012     | *10-303               |
| DDFNC  | = ***** GX | 9-228                 |
| DUMMY  | 000026 R   | #7-119                |
| EXRON  | = ***** GX | 10-304                |
| ISSAS  | = *****    | 5-1 7-115 9-235 9-248 |
| KMCL   | 000302 R   | 9-237 #9-249          |
| LF.ACT | = 100000   | #6-77 8-154           |
| LF.BRO | = 000400   | #6-77 10-277          |
| LF.BWT | = 000007   | #6-77                 |
| LF.ENA | = 002000   | #6-77                 |
| LF.LPB | = 001000   | #6-77                 |
| LF.MDC | = 000100   | #6-77                 |
| LF.MFL | = 004000   | #6-77                 |
| LF.MTP | = 000020   | #6-77 10-279          |
| LF.PAC | = 000200   | #6-77                 |
| LF.RDY | = 040000   | #6-77                 |
| LF.REA | = 010000   | #6-77                 |
| LF.SER | = 000040   | #6-77 10-281          |
| LF.TIM | = 000010   | #6-77                 |
| LF.UNL | = 020000   | #6-77                 |
| LF.X2P | = 000000   | #6-77                 |
| LN.CLO | = 000000   | #6-77                 |
| LN.DUM | = 000005   | #6-77                 |
| LN.LOA | = 000004   | #6-77                 |
| LN.LOO | = 000003   | #6-77                 |
| LN.OAU | = 000003   | #6-77                 |
| LN.OFF | = 000001   | #6-77                 |
| LN.ON  | = 000000   | #6-77                 |
| LN.OOP | = 000004   | #6-77                 |
| LN.OPE | = 000001   | #6-77                 |
| LN.REF | = 000002   | #6-77                 |
| LN.SER | = 000002   | #6-77                 |
| LN.STA | = 000017   | #6-77                 |
| LN.SUB | = 000360   | #6-77                 |
| LN.TRI | = 000006   | #6-77                 |
| L.COST | 000015     | #6-77                 |
| L.CTL  | 000012     | #6-77                 |
| L.CVA  | 177776     | #6-77                 |
| L.DDM  | 000002     | #6-77 9-227           |
| L.DDS  | 000004     | #6-77 9-226           |
| L.DLC  | 000003     | #6-77                 |
| L.DLM  | 000006     | #6-77                 |
| L.DLS  | 000010     | #6-77                 |

```

254 .SBTTL NETWORK MANAGEMENT COUNTER COMPLETION
255
256
257
258
259
260
261
262
263
264
265
266
267 000034 054134 003310 NETACP: .RAD50 /NETACP/
268
269 000040 010403 NMCMF: MOV R4,R3 ; Copy the ccb address
270 000042 062703 ADD #C.ADD,R3 ; Point to the task name
271 000046 005713 TST (R3) ; Is it for NETACP ?
272 000050 001041 BNE 40$; If NE, no
273
274 000052 116403 000030 MOV B C.CNT2(R4),R3 ; Copy SLN
275 000056 006303 ASL R3 ; Form word index
276 000060 067703 000000G ADD @SLTMA,R3 ; Point to line table address
277 000064 032773 000400 000000 BIT #LF.BRO,@(R3) ; Is it a broadcast channel?
278 000072 001026 BNE 30$; If NE, yes - EPM already filtered it.
279 000074 032773 000020 000000 BIT #LF.MTP,@(R3) ; Is it multipoint?
280 000102 001005 BNE 5$; If NE, yes - check station table
281 000104 032773 000040 000000 BIT #LF.SER,@(R3) ; Is service disabled ?
282 000112 001416 BEQ 30$; If EQ, no - allow request
283 000114 000437 BR 60$; Else, toss the request
284 000116 011303 5$: MOV (R3),R3 ; Get to the end of the line table
285 000120 062703 000022 ADD #L.LEN,R3 ; ...
286 000124 116401 000031 MOV B C.CNT2+1(R4),R1 ; Get the station number
287 000130 001404 BEQ 20$
288 000132 062703 000004 10$: ADD #S.LEN,R3 ; Move to the correct station table
289 000136 SOB R1,10$
290 000142 032713 000001 20$: BIT #SF.SER,(R3) ; Is service disabled ?
291 000146 001022 BNE 60$; If NE, yes - toss the request
292
293 000150 012703 000034' 30$: MOV #NETACP,R3 ; Else start up NETACP
294 000154 40$: CALL @SRSTD ; Scan STD for task's TCB
295 000160 103415 BCS 60$; If CS, not there !!
296
297 000162 017703 000000G MOV @NMCL2,R3 ; Get address of net man listhead
298 000166 010413 MOV R4,(R3) ; Add CCB to end of head pointer
299 000170 010477 000000G 50$: MOV R4,@NMCL2 ; Update the tail pointer
300 000174 010403 MOV R4,R3 ; Copy possible 'last' buffer address
301 000176 011404 MOV (R4),R4 ; Else, get to the end of this chain
302 000200 001373 BNE 50$; If NE, get next buffer
303 000202 052763 040000 000012 BIS #CS.LST,C.STS(R3) ; Else, set end of chain flag
304 000210 CALL @EXRQN ; Request the task (unstop or request)
305
306 000214 122764 000002 000003 60$: CMPB #CB.CCB,C.BID(R4) ; Is it a CCB ?
307 000222 001002 BNE 70$; If NE, no
308 000224 CALL @CCBRT ; Return the CCB
309 000230 SAVRG <(R4)> ; Save the next in the chain
310 000232 CALL @RDBRT ; Return this one

```

```

122 .IF DF P$$$RFL
123 .SBTTL POWERFAIL RECOVERY ROUTINE
124
125 +**--PWFAIL-POWERFAIL RECOVERY ROUTINE
126
127 THIS ROUTINE IS INVOKED ONCE PER SECOND BY THE TIMER SERVICE
128 CODE. IF THE POWERFAIL RECOVERY FLAG IS SET, WE WILL SCAN THE
129 SYSTEM LINE TABLE FOR 'ACTIVE' LINES (I.E. LINES WHICH HAVE
130 BOTH DLC AND DDM PROCESSES LOADED) AND ASYNCHRONOUSLY QUEUE
131 A CONTROL COMPLETION TO THE LLC LEVEL INDICATING THAT THE
132 LINK HAS BEEN DISCONNECTED.
133
134
135
136 000034 017701 000000G PWFAIL: MOV @PWRF1,R1 ; GET # OF LINES REMAINING TO BE POWERFAILED
137 000040 003451 ; BLE 100$; IF NONE ... NO RECOVERY UNDERWAY
138 000042 005301 ; DEC R1 ; CONVERT TO SYSTEM LINE #
139 000044 010103 ; MOV R1,R3 ; SAVE FOR LATER CALL TO $ASCMP
140
141 .IF DF N$$$1LN
142
143 MOV @SLTMA,R1 ; GET ADDRESS OF SYSTEM LINE TABLE
144 MOV (R1),R1 ; ...
145
146 .IFF
147
148 000046 006301 ASL R1 ; FORM WORD INDEX
149 000050 067701 000000G ADD @SLTMA,R1 ; POINT INTO SYSTEM LINE MAPPING TABLE
150 000054 011101 MOV (R1),R1 ; GET ADDRESS OF SYSTEM LINE TABLE
151
152 .ENDC
153
154 000056 032711 100000 BIT #LF.ACT,(R1) ; IS THIS LINE 'ACTIVE'?
155 000062 001435 BEQ 20$; NO ... NO RECOVERY REQUIRED
156 000064 005002 CLR R2 ; ASSUME LINE IS NOT MULTIPOINT
157 000066 105761 000014 TSTB L.NSTA(R1) ; IS THIS LINE MULTIPOINT?
158 000072 001413 BEQ 10$; IF EQ, NO
159 000074 116702 000067 MOVB TRIB+1,R2 ; GET TRIBUTARY NUMBER TO CHECK
160 000100 006302 ASL R2 ; FORM DOUBLE WORD OFFSET
161 000102 006302 ASL R2
162 000104 060102 ADD R1,R2 ; POINT INTO SYSTEM LINE TABLE
163 000106 132762 000200 000022 BITB #SF.ACT,L.MPF(R2) ; IS THE TRIBUTARY ACTIVE?
164 000114 001410 BEQ 15$; NO ... NO RECOVERY REQUIRED
165 000116 016702 000044 MOV TRIB,R2 ; YES ... GET TRIBUTARY ADDRESS
166 000122 150302 10$: BISE R3,R2 ; SET SYSTEM LINE NUMBER
167 000124 012703 100366 MOV #CE.DIS,R3 ; YES ... SET UP ERROR CODE
168 000130 CALL @ASCMP ; PERFORM ASYNCHRONOUS COMPLETION
169 000134 103413 BCS 100$; TRY LATER ON RESOURCE ALLOCATION FAILURE
170 000136 105267 000025 15$: INCB TRIB+1 ; UPDATE TRIBUTARY ADDRESS
171 000142 126761 000021 000014 CMPB TRIB+1,L.NSTA(R1) ; HAVE WE CHECKED ALL TRIBUTARIES ON THIS LINE?
172 000150 103731 BLO PWFAIL ; IF LO, NO
173 000152 005067 000010 CLR TRIB ; RESET TRIBUTARY ADDRESS
174 000156 005377 000000G 20$: DEC @PWRF1 ; ONE LESS LINE TO RECOVER
175 000162 001324 BNE PWFAIL ; LOOP TILL ALL DONE
176 000164 100$:
177 .IF DF X$$$MDC
178

```

CESUB1 MACRO V05.03b Friday 28-Jun-85 18:20 Page 8-1  
\$CNV18 - CONVERT TO 18-BIT UNIBUS ADDRESS

I 1

199 000142 005502  
200 000144 010203  
201 000146 012602  
202 000150  
203

10\$:

ADC R2  
MOV R2,R3  
MOV (SP)+,R2  
RETURN  
.DSABL LSB

; ADD OVERFLOW TO BITS 16 & 17  
; GET HIGH ORDER BITS INTO R3  
; AND LOW ORDER BITS INTO R2  
; RETURN

CESUB1      CREATED BY MACRO ON 28-JUN-85 AT 18:21      PAGE 2      I 2

SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL  | VALUE      | REFERENCES                                 |
|---------|------------|--------------------------------------------|
| \$MVFBF | 000216 RG  | #10-293                                    |
| \$MVTBF | 000152 RG  | #9-244                                     |
| \$PDVID | 000460 RG  | #13-430                                    |
| \$PDVNM | = ***** GX | 13-432                                     |
| \$PDVTA | = ***** GX | 11-335      11-348      13-431      13-443 |
| \$PUMR  | = ***** GX | 8-181                                      |
| \$XBIAS | = ***** GX | 12-390                                     |

\*\*FILE\*\*ID\*\*CEXCM

```

CCCCCCCC EEEEEEEEE XX XX CCCCCCCC MM MM
CCCCCCCC EEEEEEEEE XX XX CCCCCCCC MM MM
CC EE XX XX CC MMMM MMMM
CC EE XX XX CC MM MM MM
CC EE XX XX CC MM MM MM
CC EEEEEEEE XX XX CC MM MM MM
CC EEEEEEEE XX XX CC MM MM MM
CC EE XX XX CC MM MM MM
CC EE XX XX CC MM MM MM
CC EE XX XX CC MM MM MM
CC EEEEEEEE XX XX CCCCCCCC MM MM
CCCCCCCC EEEEEEEEE XX XX CCCCCCCC MM MM
CCCCCCCC EEEEEEEEE XX XX CCCCCCCC MM MM

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```

\*\*FILE\*\*ID\*\*STCRC

I 4

```
SSSSSSSS TTTTTTTTT CCCCCCCC RRRRRRRR CCCCCCCC
SSSSSSSS TTTTTTTTT CCCCCCCC RRRRRRRR CCCCCCCC
SS TT CC RR CC
SS TT CC RR CC
SS TT CC RR CC
SS TT CC RR CC
SSSSSS TT CC RRRRRRRR CC
SSSSSS TT CC RRRRRRRR CC
SS TT CC RR CC
SS TT CC RR CC
SS TT CC RR CC
SS TT CC RR CC
SSSSSSSS TT CCCCCCCC RR RR CCCCCCCC
SSSSSSSS TT CCCCCCCC RR RR CCCCCCCC
```

....  
....  
....  
....

```
LL SSSSSSSS TTTTTTTTT
LL SSSSSSSS TTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLL SSSSSSSS TT
LLLLLLLLL SSSSSSSS TT
```



```

390 MOV CTABL-.(R5),R5 ; GET FIRST MODIFIER WORD
391
392 .ENDC
393
394 BIC #^C<360>,R4 ; NOW EXTRACT HIGH 4 BITS
395 ASR R4 ; FORM WORD OFFSET
396 ASR R4 ; ...
397 ASR R4 ; ...
398
399 .IF DF M$$MGE
400
401 MOV CTABL+32.(R4),R4; GET SECOND MODIFIER WORD
402
403 .IFF
404
405 ADD PC,R4 ; FORM PIC ADDRESS
406 MOV CTABL+32-.(R4),R4 ; GET SECOND MODIFIER WORD
407
408 .ENDC
409
410 XOR R4,R5 ; EXCLUSIVE 'OR' MODIFIER WORDS
411 CLKB R1 ; CLEAR LOW BYTE OF OLD CRC
412 SWAB R1 ; NOW WORK ON UPPER BYTE
413 XOR R5,R1 ; PRODUCE UPDATED CRC
414
415 .ENDC
416
417 DEC R0 ; ANY MORE TO GO?
418 BGT 10$; IF GT, YES
419 20$:
420 .IF NDF R$$EIS
421
422 RESRG <R3>
423
424 .ENDC
425
426 .ENDC
427
428 RESRG <R5,R4,R2,R0>
429 RETURN

```

STCRCF MACRO V05.03b Friday 28-Jun-85 18:22

Table of contents

|     |     |                                                |
|-----|-----|------------------------------------------------|
| 6-  | 63  | MACRO DEFINITIONS                              |
| 7-  | 90  | DEFINE KG-11 REGISTERS AND BITS                |
| 8-  | 101 | EXECUTIVE VECTOR TABLE                         |
| 9-  | 109 | \$CLCRC - CALCULATE CRC-16 ON A TRANSMIT CHAIN |
| 10- | 215 | \$STCRC - CALCULATE CRC ON BLOCK OF DATA       |
| 11- | 276 | STCR2 - CALCULATE CRC                          |
| 12- | 433 | MODIFIER TABLE FOR SOFTWARE CRC                |

```

431 .IF NDF K$$$G11
432
433 .SBTTL MODIFIER TABLE FOR SOFTWARE CRC
434
435
436 000234 000000 CTABL: .WORD 0 ;FIRST HALF OF TABLE
437 000236 140301 .WORD 140301
438 000240 140601 .WORD 140601
439 000242 000500 .WORD 500
440 000244 141401 .WORD 141401
441 000246 001700 .WORD 1700
442 000250 001200 .WORD 1200
443 000252 141101 .WORD 141101
444 000254 143001 .WORD 143001
445 000256 003300 .WORD 3300
446 000260 003600 .WORD 3600
447 000262 143501 .WORD 143501
448 000264 002400 .WORD 2400
449 000266 142701 .WORD 142701
450 000270 142201 .WORD 142201
451 000272 002100 .WORD 2100
452
453 .IF DF F$$$AST
454
455 000274 146001 .WORD 146001
456 000276 006300 .WORD 6300
457 000300 006600 .WORD 6600
458 000302 146501 .WORD 146501
459 000304 007400 .WORD 7400
460 000306 147701 .WORD 147701
461 000310 147201 .WORD 147201
462 000312 007100 .WORD 7100
463 000314 005000 .WORD 5000
464 000316 145301 .WORD 145301
465 000320 145601 .WORD 145601
466 000322 005500 .WORD 5500
467 000324 144401 .WORD 144401
468 000326 004700 .WORD 4700
469 000330 004200 .WORD 4200
470 000332 144101 .WORD 144101
471 000334 154001 .WORD 154001
472 000336 014300 .WORD 14300
473 000340 014600 .WORD 14600
474 000342 154501 .WORD 154501
475 000344 015400 .WORD 15400
476 000346 155701 .WORD 155701
477 000350 155201 .WORD 155201
478 000352 015100 .WORD 15100
479 000354 017000 .WORD 17000
480 000356 157301 .WORD 157301
481 000360 157601 .WORD 157601
482 000362 017500 .WORD 17500
483 000364 156401 .WORD 156401
484 000366 016700 .WORD 16700
485 000370 016200 .WORD 16200
486 000372 156101 .WORD 156101
487 000374 012000 .WORD 12000

```

```
1 .IF DF K$$G11
2 .TITLE STCRCK
3 .IFF
4 .IF DF F$$AST
5 .TITLE STCRCK
6 .IFF
7 .TITLE STCRCK
8 .ENDC
9 .ENDC
10 .IDENT /V05.00/
```

```
11
12 *
13 COPYRIGHT (C) 1978,1979,1980, 1982, 1983, 1985 BY
14 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
```

```
15 THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
16 SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
17 INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR
18 ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE
19 MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH
20 SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE
21 TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN
22 IN DEC.
```

```
23 THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
24 NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
25 EQUIPMENT CORPORATION.
```

```
26 DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
27 ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
```

```
28 MODULE DESCRIPTION
```

```
29 CRC-16 CALCULATION ROUTINES
```

```
30 DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
```

```
31 IDENT HISTORY:
```

- ```
32     1.00  10-FEB-78
33           VERSION 2.0 RELEASE
34
35     2.00  14-DEC-79
36           DECNET-11M/S V3.0
37           DECNET-11M-PLUS V1.0
38
39     3.00  16-APR-82
40           DECNET-11M V3.1
41           DECNET-11M-PLUS V1.1
42
43     4.00  07-NOV-83
44           DECNET-11M V4.0
45           DECNET-11M-PLUS V2.0
46
47     5.00  22-JUL-85
48           DECnet-11M/S V4.2
49           DECnet-11M-Plus V3.0
50
51
52
53
54
55
56
57
```

488	.WORD	152301
489	.WORD	152601
490	.WORD	12500
491	.WORD	153401
492	.WORD	13700
493	.WORD	13200
494	.WORD	153101
495	.WORD	151001
496	.WORD	11300
497	.WORD	11600
498	.WORD	151501
499	.WORD	10400
500	.WORD	150701
501	.WORD	150201
502	.WORD	10100
503	.WORD	170001
504	.WORD	30300
505	.WORD	30600
506	.WORD	170501
507	.WORD	31400
508	.WORD	171701
509	.WORD	171201
510	.WORD	31100
511	.WORD	33000
512	.WORD	173301
513	.WORD	173601
514	.WORD	33500
515	.WORD	172401
516	.WORD	32700
517	.WORD	32200
518	.WORD	172101
519	.WORD	36000
520	.WORD	176301
521	.WORD	176601
522	.WORD	36500
523	.WORD	177401
524	.WORD	37700
525	.WORD	37200
526	.WORD	177101
527	.WORD	175001
528	.WORD	35300
529	.WORD	35600
530	.WORD	175501
531	.WORD	34400
532	.WORD	174701
533	.WORD	174201
534	.WORD	34100
535	.WORD	24000
536	.WORD	164301
537	.WORD	164601
538	.WORD	24500
539	.WORD	165401
540	.WORD	25700
541	.WORD	25200
542	.WORD	165101
543	.WORD	167001
544	.WORD	27300

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

.TITLE AXBFR
.IDENT /V05.00/

COPYRIGHT (C) 1978,1979,1980, 1982, 1983, 1985 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION
AUX BUFFER WAIT COMPLETION PROCESSOR

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 10-FEB-78
VERSION 2.0 RELEASE
- 2.00 14-DEC-79
DECNET-11M/S V3.0
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82
DECNET-11M V3.1
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83
DECNET-11M V4.0
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85
DECnet-11M/S V4.2
DECnet-11M-Plus V3.0
DECnet-Micro/RSX V1.0

```

42
43      ; Vector table to support vectored executive
44
45 000000 000000      $AUXVT::WORD 0      ; Start of table
46
47      ; RSX executive symbols
48
49 000002 000000G     KSAR6::WORD KISAR6
50 000004 000000G     ALOCB::WORD $ALOCB
51
52 000006 000000G     CLINS::WORD $CLINS
53
54      .IF DF R$$MPL
55      .IF NDF R$$PRO
56 CPURM::WORD $CPURM
57      .ENDC
58      .ENDC
59
60 000010 000000G     DEACB::WORD $DEACB
61 000012 000000G     EXRQF::WORD $EXRQF
62 000014 000000G     EXRQN::WORD $EXRQN
63 000016      FMSK2::WORD $FMSK2
64      .IF NDF R$$PRO
65      .WORD $FMSK2
66      .IFF ; NDF R$$PRO
67      .WORD $FMSK+2
68      .ENDC ; NDF R$$PRO
69      .IFF ; DF R$$MPL
70 000016 000002G     .WORD $FMSK+2
71      .ENDC ; DF R$$MPL
72 INTCT::WORD $INTCT
73 SRSTD::WORD $SRSTD
74 TKPS::WORD $TKPS
75 TSKRT::WORD $TSKRT
76 UMRPT::WORD $UMRPT
77
78      .IF DF R$$MPL
79      .IF NDF R$$PRO
80 .1SCTB::WORD $1SCTB
81      .ENDC
82      .ENDC
83
84      ;
85      ; Communication executive symbols
86
86 000032 000000G     ASCMP::WORD $ASCMP
87 000034 000000G     CCBGT::WORD $CCBGT
88 000036 000000G     CCBRT::WORD $CCBRT
89 000040 000000G     CEACC::WORD $CEACC
90 000042 000000G     CEDIV::WORD $CEDIV
91 000044 000000G     CMFRK::WORD $CMFRK
92 000046 000000G     CXOPT::WORD $CXOPT
93 000050 000000G     DDAST::WORD $DDAST
94 000052 000000G     DDCCP::WORD $DDCCP
95 000054 000000G     DDFNC::WORD $DDFNC
96 000056 000000G     DDMSN::WORD $DDMSN
97 000060 000000G     DSPTM::WORD $DSPTM
98

```

```

122      .IF DF P$$$RFL
123      .SBTTL POWERFAIL RECOVERY ROUTINE
124
125      +
126      ***-PWFAIL-POWERFAIL RECOVERY ROUTINE
127
128      THIS ROUTINE IS INVOKED ONCE PER SECOND BY THE TIMER SERVICE
129      CODE. IF THE POWERFAIL RECOVERY FLAG IS SET, WE WILL SCAN THE
130      SYSTEM LINE TABLE FOR 'ACTIVE' LINES (I.E. LINES WHICH HAVE
131      BOTH DLC AND DDM PROCESSES LOADED) AND ASYNCHRONOUSLY QUEUE
132      A CONTROL COMPLETION TO THE LLC LEVEL INDICATING THAT THE
133      LINK HAS BEEN DISCONNECTED.
134
135      -
136
137      PWFAIL: MOV    @PWRF1,R1      ; GET # OF LINES REMAINING TO BE POWERFAILED
138              BLE    100$          ; IF NONE ... NO RECOVERY UNDERWAY
139              DEC     R1            ; CONVERT TO SYSTEM LINE #
140              MOV     R1,R3        ; SAVE FOR LATER CALL TO $ASCMP
141
142      .IF DF N$$$1LN
143
144      MOV     @SLTMA,R1          ; GET ADDRESS OF SYSTEM LINE TABLE
145      MOV     (R1),R1           ; ...
146
147      .IFF
148
149      ASL     R1                ; FORM WORD INDEX
150      ADD     @SLTMA,R1         ; POINT INTO SYSTEM LINE MAPPING TABLE
151      MOV     (R1),R1          ; GET ADDRESS OF SYSTEM LINE TABLE
152
153      .ENDC
154
155      BIT     #LF.ACT,(R1)      ; IS THIS LINE 'ACTIVE'?
156      BEQ     20$              ; NO ... NO RECOVERY REQUIRED
157      CLR     R2                ; ASSUME LINE IS NOT MULTIPOINT
158      TSTB    L.NSTA(R1)       ; IS THIS LINE MULTIPOINT?
159      BEQ     10$              ; IF EQ, NO
160      MOV     TRIB+1,R2         ; GET TRIBUTARY NUMBER TO CHECK
161      ASL     R2                ; FORM DOUBLE WORD OFFSET
162      ADD     R1,R2             ; POINT INTO SYSTEM LINE TABLE
163      BIT     #SF.ACT,L.MPF(R2) ; IS THE TRIBUTARY ACTIVE?
164      BEQ     15$              ; NO ... NO RECOVERY REQUIRED
165      MOV     TRIB,R2           ; YES ... GET TRIBUTARY ADDRESS
166      BISB    R3,R2            ; SET SYSTEM LINE NUMBER
167      MOV     #CE.DIS,R3        ; YES ... SET UP ERROR CODE
168      CALL    @ASCMP           ; PERFORM ASYNCHRONOUS COMPLETION
169      BCS     100$             ; TRY LATER ON RESOURCE ALLOCATION FAILURE
170      INCB    TRIB+1           ; UPDATE TRIBUTARY ADDRESS
171      CMPB    TRIB+1,L.NSTA(R1) ; HAVE WE CHECKED ALL TRIBUTARIES ON THIS LINE?
172      BLO     PWFAIL           ; IF LO, NO
173      CLR     TRIB             ; RESET TRIBUTARY ADDRESS
174      DEC     @PWRF1           ; ONE LESS LINE TO RECOVER
175      BNE     PWFAIL           ; LOOP TILL ALL DONE
176
177      100$:
178      .IF DF X$$$MDC

```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```
.IF DF R$$$1D!$$$AS
.IF DF X$$$MDC
.TITLE AXDSPM - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSP - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.ENDC
.IFF
.IF DF X$$$MDC
.IF DF P$$$RFL
.TITLE AXDSPB - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSPM - AUXILLIARY PROCESS DISPATCH
.ENDC
.IFF
.IF DF P$$$RFL
.TITLE AXDSP - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSP - AUXILLIARY PROCESS DISPATCH
.ENDC
.ENDC
.IDENT /V05.01/
```

```
:
: COPYRIGHT (C) 1978,1979,1980, 1983, 1985 B"
: DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
:
: THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
: ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
: INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
: COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
: OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
: TRANSFERRED.
:
: THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
: AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
: CORPORATION.
:
: DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
: SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
:
: MODULE DESCRIPTION
:
:     AUXILIARY PROCESS DISPATCH TABLE
:
: DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
:
: IDENT HISTORY:
: 1.00 10-FEB-78
:     VERSION 2.0 RELEASE
:
: 2.00 14-DEC-79
```

AXDSPB CREATED BY MACRO ON 3-SEP-85 AT 10:58

PAGE 2 I 14

SYMBOL CROSS REFERENCE

CREF 04.00

SYMBOL	VALUE	REFERENCES
L.FLG	000000	#6-77
L.KRBA	000016	#6-77
L.LEN	= 000022	#6-77 10-285
L.MPF	000022	#6-77 8-163
L.NMST	000020	#6-77
L.NSTA	000014	#6-77 8-157 8-171
L.OWNR	000021	#6-77
L.UNT	000013	#6-77
MDMCTL	= ***** GX	7-89
MDMSCN	= ***** GX	8-179
NETACP	= 000306 R	#10-267 10-293
NMCL2	= ***** GX	10-297 10-299
NMCMP	= 000312 R	7-112 #10-269
N\$\$ILN	= *****	8-141
PDSPL	= ***** GX	9-229
PWFAIL	= 000034 R	7-91 #8-136 8-172 8-175
PWRFL	= 000172 R	7-108 #9-202
PWRFL	= ***** GX	8-136 8-174 9-202 9-245
P\$\$RFL	= 000001	#4-2 5-9 7-90 7-107 8-122
RDBRT	= ***** GX	10-310
R\$\$MPL	= *****	9-213
R\$\$11D	= *****	5-1 7-115
SF.ACT	= 000200	#6-77 8-163
SF.ENA	= 000100	#6-77
SF.LPB	= 000004	#6-77
SF.MFL	= 000040	#6-77
SF.PAC	= 000020	#6-77
SF.REA	= 000010	#6-77
SF.SER	= 000001	#6-77 10-290
SF.SVC	= 000002	#6-77
SF.UNL	= 000040	#6-77
SLTMA	= ***** GX	8-149 9-208 10-276
SLTNF	= ***** GX	9-203 9-245
SRSTD	= ***** GX	9-238 10-294
S.COST	= 000001	#6-77
S.FLG	= 000000	#6-77
S.LEN	= 000004	#6-77 10-288
S.NMST	= 000002	#6-77
S.OWNR	= 000003	#6-77
TRIB	= 000170 R	8-159 8-165 *8-170 8-171 *8-173 #8-191
TSKRT	= ***** GX	9-241
X\$\$MDC	= 000001	#4-3 5-8 7-88 7-115 8-177
\$AUXTB	= 000000 RG	#7-84
\$BFRTN	= ***** GX	7-86

AXDSPM - AUXILIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 10:59 Page 10-1
NETWORK MANAGEMENT COUNTER COMPLETION

311	000236		RESRG	<R4>	; Recover the next buffer
312	000240	001373	BNE	70\$; If NE, get next buffer
313	000242		RETURN		; Else, return
314					
315	000001		.END		

```

179 CALLR MDMSCN ; MODEM CONTROL SCAN ROUTINE
180
181 .IFF
182
183 000164 RETURN
184
185 .ENDC
186
187 ;+
188 ; LOCAL STORAGE FOR TRIBUTARY ADDRESS DURING POWERFAIL
189 ; -
190
191 000166 TRIB: .BLKW

```

```

205 .SBTTL $MVTBF - MOVE FROM MAPPED BUFFER TO UNMAPPED BUFFER
206
207
208 *** $MVTBF - MOVE FROM MAPPED BUFFER TO UNMAPPED BUFFER
209
210 THIS ROUTINE IS CALLED TO MOVE A BLOCK OF MEMORY (LESS THAN
211 4K WORDS) FROM A BUFFER THAT IS CURRENTLY MAPPED (EITHER
212 IN SYSTEM DYNAMIC SPACE OR IN THE COMM BUFFER POOL) TO AN
213 UNMAPPED BUFFER. NOTE THE MAPPED BUFFER CAN NOT BE IN THE PROCESS'S
214 SPACE MAPPED VIA APR5 SINCE IT USES APR5 TO MAP TO THE UNMAPPED
215 BUFFER.
216
217 CALLING FORMAT:
218 JSR R1,$MVTBF
219
220 INPUTS:
221 R2 = VIRTUAL ADDRESS OF MAPPED 'FROM' BUFFER
222 R3 = NUMBER OF BYTES TO MOVE
223 ON THE STACK:
224
225     R1      ORIGINAL CONTENTS OF R1 BEFORE CALL
226     VA      16-BIT VIRTUAL ADDRESS OF UNMAPPED 'TO' BFFER
227     BIAS    RELOCATION BIAS OF UNMAPPED 'TO' BUFFER
228             (MAPPED SYSTEMS ONLY)
229
230
231 OUTPUTS:
232 R2    UPDATED ADDRESS OF 'FROM' BUFFER
233       POINTS TO LAST BYTE MOVED +1
234 R3 =   ZERO
235
236       STILL MAPPED TO 'FROM' BUFFER VIA KISAR6
237
238 REGISTERS MODIFIED:
239 R2 & R3
240
241
242
243
244 .ENABL LSB
245 $MVTBF::
246 000152 016746 000000G 000000G MOV KISAR5, -(SP) ; SAVE CURRENT PROCESS MAPPING
247 000156 016667 000006 000000G MOV 6(SP), KISAR5 ; MAP TO 'TO' BUFFER
248 000164 010166 000006 000000G MOV R1, 6(SP) ; SAVE RETURN ADDRESS
249 000170 016601 000004 000000G MOV 4(SP), R1 ; GET 'TO' BUFFER VIRTUAL ADDRESS
250 000174 022701 140000 000000G CMP #140000, R1 ; IS THE 'TO' BUFFER IN THE EXEC POOL?
251 000200 101002 000000 000000G BHI 10$ ; IF HI, YES THEN DON'T ALTER THE VIRTUAL ADDRESS
252 000202 162701 020000 000000G SUB #20000, R1 ; SET VA FOR BIAS VIA KISAR5
253 000206 112221 000000 000000G 10$: MOVB (R2)+, (R1)+ ; MOVE BUFFER A BYTE AT A TIME
254 000210 000421 000000 000000G SOB R3, 10$ ; LOOP TILL DONE
255 000214 000421 000000 000000G BR 20$ ; JOIN COMMON EXIT CODE

```

CESUB1 CREATED BY MACRO ON 28-JUN-85 AT 18:21

PAGE 3 J 2

MACRO CROSS REFERENCE

CREF 04.00

MACRO NAME REFERENCES

CALL	11-355							
CALLR	#5-62							
CCBDF\$	#5-61	5-63						
CLKDF\$	#5-61	5-65						
ENABL\$	#5-60							
INHIB\$	#5-60							
PDVDF\$	#5-61	5-64						
RESRG	#5-60	6-98	7-139	11-353	12-412	13-446		
RETURN	6-99	7-140	8-202	10-306	11-363	12-413		
SAVRG	#5-60	6-90	7-128	11-334	12-393	13-430	13-447	14-473
SOB	9-2	10-302						15-507

```
.IIF DF R$$MPL .TITLE NETCM - CEX COMMON DATABASE
.IIF NDF R$$MPL .TITLE CEXCM - CEX COMMON DATABASE
.IDENT /V05.00/
```

```
.....
COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
```

```
.....
THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.
```

```
.....
THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.
```

```
.....
DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
```

```
.....
MODULE DESCRIPTION:
```

```
.....
COMMUNICATIONS EXECUTIVE COMMON DATABASE
```

```
.....
IDENT HISTORY:
```

```
.....
1.00 14-DEC-79
      DECNET-11M/S V3.0
      DECNET-11M-PLUS V1.0
.....
3.00 16-APR-82
      DECNET-11M V3.1
      DECNET-11M-PLUS V1.1
.....
4.00 07-NOV-83
      DECNET-11M V4.0
      DECNET-11M-PLUS V2.0
.....
5.00 22-JUL-85
      DECnet-11M/S V4.2
      DECnet-11M-Plus V3.0
      DECnet-Micro/RSX V1.0
.....
```

```
.....
.IF DF M$$NET
```

```
.....
.IF NDF R$$MPL
```

```
.....
.MCALL CCBDF$,NKRDF$,OPTDF$
```

```
.....
CCBDF$ ; DEFINE CCB OFFSETS
NKRDF$ <:;>,<=> ; DEFINE NETWORK KRB OFFSETS
```

```
56 000000
57 000000
```

STCRC MACRO V05.03b Friday 28-Jun-85 18:21
Table of contents

5-	63	MACRO DEFINITIONS
6-	90	DEFINE KG-11 REGISTERS AND BITS
7-	101	EXECUTIVE VECTOR TABLE
8-	109	\$CLCRC - CALCULATE CRC-16 ON A TRANSMIT CHAIN
9-	215	\$STCRC - CALCULATE CRC ON BLOCK OF DATA
10-	276	STCR2 - CALCULATE CRC
11-	433	MODIFIER TABLE FOR SOFTWARE CRC


```

431 .IF NDF K$$G11
432
433 .SBTTL MODIFIER TABLE FOR SOFTWARE CRC
434
435
436 000256 000000 CTABL: .WORD 0 ;FIRST HALF OF TABLE
437 000260 140301 .WORD 140301
438 000262 140601 .WORD 140601
439 000264 000500 .WORD 500
440 000266 141401 .WORD 141401
441 000270 001700 .WORD 1700
442 000272 001200 .WORD 1200
443 000274 141101 .WORD 141101
444 000276 143001 .WORD 143001
445 000300 003300 .WORD 3300
446 000302 003600 .WORD 3600
447 000304 143501 .WORD 143501
448 000306 002400 .WORD 2400
449 000310 142701 .WORD 142701
450 000312 142201 .WORD 142201
451 000314 002100 .WORD 2100
452
453 .IF DF F$$AST
454
455 .WORD 146001
456 .WORD 6300
457 .WORD 6600
458 .WORD 146501
459 .WORD 7400
460 .WORD 147701
461 .WORD 147201
462 .WORD 7100
463 .WORD 5000
464 .WORD 145301
465 .WORD 145601
466 .WORD 5500
467 .WORD 144401
468 .WORD 4700
469 .WORD 4200
470 .WORD 144101
471 .WORD 154001
472 .WORD 14300
473 .WORD 14600
474 .WORD 154501
475 .WORD 15400
476 .WORD 155701
477 .WORD 155201
478 .WORD 15100
479 .WORD 17000
480 .WORD 157301
481 .WORD 157601
482 .WORD 17500
483 .WORD 156401
484 .WORD 16700
485 .WORD 16200
486 .WORD 156101
487 .WORD 12000

```

```
.IF DF K$$G11
.TITLE STCRCK
.IFF
.IF DF F$$AST
.TITLE STCRCF
.IFF
.TITLE STCRC
.ENDC
.ENDC
.IDENT /V05.00/
```

```

+
: COPYRIGHT (C) 1978,1979,1980, 1982, 1983, 1985 BY
: DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
```

```

: THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
: SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
: INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR
: ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE
: MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH
: SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE
: TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN
: IN DEC.
```

```

: THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
: NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
: EQUIPMENT CORPORATION.
```

```

: DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
: ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
```

MODULE DESCRIPTION

CRC-16 CALCULATION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 10-FEB-78
VERSION 2.0 RELEASE
- 2.00 14-DEC-79
DECNET-11M/S V3.0
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82
DECNET-11M V3.1
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83
DECNET-11M V4.0
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85
DECnet-11M/s V4.2
DECnet-11M-Plus V3.0

488	000376	152301	.WORD	152301
489	000400	152601	.WORD	152601
490	000402	012500	.WORD	12500
491	000404	153401	.WORD	153401
492	000406	013700	.WORD	13700
493	000410	013200	.WORD	13200
494	000412	153101	.WORD	153101
495	000414	151001	.WORD	151001
496	000416	011300	.WORD	11300
497	000420	011600	.WORD	11600
498	000422	151501	.WORD	151501
499	000424	010400	.WORD	10400
500	000426	150701	.WORD	150701
501	000430	150201	.WORD	150201
502	000432	010100	.WORD	10100
503	000434	170001	.WORD	170001
504	000436	030300	.WORD	30300
505	000440	030600	.WORD	30600
506	000442	170501	.WORD	170501
507	000444	031400	.WORD	31400
508	000446	171701	.WORD	171701
509	000450	171201	.WORD	171201
510	000452	031100	.WORD	31100
511	000454	033000	.WORD	33000
512	000456	173301	.WORD	173301
513	000460	173601	.WORD	173601
514	000462	033500	.WORD	33500
515	000464	172401	.WORD	172401
516	000466	032700	.WORD	32700
517	000470	032200	.WORD	32200
518	000472	172101	.WORD	172101
519	000474	036000	.WORD	36000
520	000476	176301	.WORD	176301
521	000500	176601	.WORD	176601
522	000502	036500	.WORD	36500
523	000504	177401	.WORD	177401
524	000506	037700	.WORD	37700
525	000510	037200	.WORD	37200
526	000512	177101	.WORD	177101
527	000514	175001	.WORD	175001
528	000516	035300	.WORD	35300
529	000520	035600	.WORD	35600
530	000522	175501	.WORD	175501
531	000524	034400	.WORD	34400
532	000526	174701	.WORD	174701
533	000530	174201	.WORD	174201
534	000532	034100	.WORD	34100
535	000534	024000	.WORD	24000
536	000536	164301	.WORD	164301
537	000540	164601	.WORD	164601
538	000542	024500	.WORD	24500
539	000544	165401	.WORD	165401
540	000546	025700	.WORD	25700
541	000550	025200	.WORD	25200
542	000552	165101	.WORD	165101
543	000554	167001	.WORD	167001
544	000556	027300	.WORD	27300

58
59
60
61

:
:
:
:

DECnet-Micro/RSX V1.0

545	.WORD	27600
546	.WORD	167501
547	.WORD	26400
548	.WORD	166701
549	.WORD	166201
550	.WORD	26100
551	.WORD	162001
552	.WORD	22300
553	.WORD	22600
554	.WORD	162501
555	.WORD	23400
556	.WORD	163701
557	.WORD	163201
558	.WORD	23100
559	.WORD	21000
560	.WORD	161301
561	.WORD	161601
562	.WORD	21500
563	.WORD	160401
564	.WORD	20700
565	.WORD	20200
566	.WORD	160101
567	.WORD	120001
568	.WORD	60300
569	.WORD	60600
570	.WORD	120501
571	.WORD	61400
572	.WORD	121701
573	.WORD	121201
574	.WORD	61100
575	.WORD	63000
576	.WORD	123301
577	.WORD	123601
578	.WORD	63500
579	.WORD	122401
580	.WORD	62700
581	.WORD	62200
582	.WORD	122101
583	.WORD	66000
584	.WORD	126301
585	.WORD	126601
586	.WORD	66500
587	.WORD	127401
588	.WORD	67700
589	.WORD	67200
590	.WORD	127101
591	.WORD	125001
592	.WORD	65300
593	.WORD	65600
594	.WORD	125501
595	.WORD	64400
596	.WORD	124701
597	.WORD	124201
598	.WORD	64100
599	.WORD	74000
600	.WORD	134301
601	.WORD	134601

AXBFR MACRO V05.03b Friday 28-Jun-85 18:28 Page 5
Macro definitions

56
57
58
59 000000
60 000000
61 000000

.SBTTL Macro definitions
.MCALL CCBDF\$,SLTDF\$,NHWDF\$,ENABL\$,INHIB\$
CCBDF\$
SLTDF\$
NHWDF\$

```

99                                     .IF DF R$$MPL
100                                    .IF NDF R$$PRO
101      M100Q:: .WORD $M100Q
102                                     .ENDC
103                                     .ENDC
104
105      000062  000000G      NMCLH:: .WORD $NMCLH
106      000064          NMCL2:: .IF DF R$$PRO
107                                     .WORD $NMCLH+2
108                                     .IFF ; DF R$$PRO
109      000064  000000G      .WORD $NMCL2
110                                     .ENDC ; DF R$$PRO
111      000066  000000G      PDDSP:: .WORD $PDDSP
112      000070  000000G      PDQUE:: .WORD $PDQUE
113      000072  000000G      PDSPL:: .WORD $PDSPL
114      000074  000000G      PDVNM:: .WORD $PDVNM
115      000076  000000G      PDVTA:: .WORD $PDVTA
116      000100  000000G      PUMR:: .WORD $PUMR
117      000102  000000G      PWRF1:: .WORD $PWRF1
118      000104  000000G      QSTAT:: .WORD $QSTAT
119      000106  000000G      RDBG1:: .WORD $RDBG1
120      000110  000000G      RDBNM:: .WORD $RDBNM
121      000112  000000G      RDBRT:: .WORD $RDBRT
122      000114  000000G      RDBSZ:: .WORD $RDBSZ
123      000116  000000G      RDBTH:: .WORD $RDBTH
124      000120  000000G      RDQCT:: .WORD $RDQCT
125      000122  000000G      RDQSL:: .WORD $RDQSL
126      000124  000000G      SLTMA:: .WORD $SLTMA
127      000126  000000G      SLTNM:: .WORD $SLTNM
128      000130  000000G      SQRCM:: .WORD $SQRCM
129      000132  000000G      STDD1:: .WORD $STDD1
130      000134  000000G      STDLC:: .WORD $STDLC
131      000136  000000G      STD11:: .WORD $STD11
132      000140  000000G      STMFC:: .WORD $STMFC
133                                     .IF DF R$$MPL
134                                    .IF NDF R$$PRO
135      STMTB:: .WORD $STMTB
136                                     .ENDC
137                                     .ENDC
138      000142  000000G      TK100:: .WORD $TK100
139      000144  000000G      TSTIM:: .WORD $TSTIM
140      000146  000000G      T1SCL:: .WORD $T1SCL
141      000150  000000G      T100C:: .WORD $T100C
142      000152  000000G      T100Q:: .WORD $T100Q
143      000154  000000G      XAVL:: .WORD $XAVL
144      000156  000000G      ZTIME:: .WORD $ZTIME
145      000160          Z1M2:: .IF
146                                     .WORD $ZTIME+2
147                                     .IFF ; DF R$$PRO
148      000160  000000G      .WORD $Z1M2
149                                     .ENDC ; DF R$$PRO
150
151                                     .IF DF R$$MPL
152      CKURM:: .WORD $CKURM
153      CPBIT:: .WORD $CPBIT
154
155                                     .IF NDF R$$PRO

```

```
179          CALLR  MDMSCN          ; MODEM CONTROL SCAN ROUTINE
180
181          .IFF
182
183          RETURN
184
185          .ENDC
186
187          ;+
188          ; LOCAL STORAGE FOR TRIBUTARY ADDRESS DURING POWERFAIL
189          ; -
190
191          TRIB:  .BLKW
```


58	:	DECNET-11M/S V3.0
59	:	DECNET-11M-PLUS V1.0
60	:	
61	:	4.00 07-NOV-83
62	:	DECNET-11M V4.0
63	:	DECNET-11M-PLUS V2.0
64	:	
65	:	5.00 22-JUL-85
66	:	DECnet-11M/S V4.2
67	:	DECnet-11M-Plus V3.0
68	:	DECnet-Micro/RSX V1.0
69	:	
70	:	5.01 09-Aug-85
71	:	Add logic to bypass service disabled check on broadcast channels.
72	:	This change requires EPMMAI ident V5.01.
73	:	

AXDSPB CREATED BY MACRO ON 3-SEP-85 AT 10:58

PAGE 3 J 14

MACRO CROSS REFERENCE

CREF 04.00

MACRO NAME REFERENCES

CALL	8-168	9-229	9-238	9-241	10-294	10-310
CALLR	7-120	8-179	10-304	10-308		
CCBDF\$	#6-75	6-78				
RESRG	#6-75	10-311				
RETURN	7-119	9-246	10-313			
SAVRG	#6-75	10-309				
SLTDF\$	#6-75	6-77				
SOB	10-289					

ASSCHK= 000000	CS.DEV= 000002	D\$SYNM= 000000	G\$SWRD= 000000	L.OWNR 000021
ASSCPS= 000000	CS.DIS= 000040	EXRQN = ***** GX	I\$SRAR= 000000	L.UNT 000013
ASSPRI= 000000	CS.ENA= 000001	E\$XPR= 000000	I\$SRDN= 000000	MDMCTL= ***** GX
ASSTRP= 000000	CS.ENB= 000020	FC.CCP= 000020	K\$CNT= 177546	MDMSCN= ***** GX
CB.CCB= 000002	CS.ERR= 100000	FC.CTL= 000006	K\$CSR= 177546	M\$CRB= 000124
CB.DDM= 000040	CS.FTL= 001000	FC.KCP= 000016	K\$SLDC= 000000	M\$CRX= 000000
CB.DLC= 000020	CS.HCR= 000001	FC.KIL= 000004	K\$STPS= 000074	M\$FCS= 000000
CB.RDB= 000004	CS.HFE= 002000	FC.MAN= 000024	LD\$LP= 000000	M\$MGE= 000000
CB.SDB= 000010	CS.LST= 040000	FC.MLD= 000026	LF.ACT= 100000	M\$NET= 000000
CB.SLI= 000100	CS.MTL= 004000	FC.PCT= 000030	LF.BRO= 000400	M\$OVR= 000000
CB.XLB= 000001	CS.RNG= 000010	FC.PWR= 000022	LF.BWT= 000007	NETACP 000034R
CCBRET 000030R	CS.ROV= 000004	FC.RCE= 000002	LF.ENA= 002000	NMCL2 = ***** GX
CCBRT = ***** GX	CS.RSN= 010000	FC.RCP= 000014	LF.LPB= 001000	NMCMR 000040R
CC.LLC= 000200	CS.SHU= 000001	FC.TIM= 000010	LF.MDC= 000100	N\$SACC= 000001
CE.ABO= 100362	CS.SID= 000002	FC.XCP= 000012	LF.MFL= 004000	N\$SBUF= 000001
CE.DAO= 100346	CS.STR= 000004	FC.XME= 000000	LF.MTP= 000020	N\$SLDV= 000001
CE.DIS= 100366	CS.SUC= 000001	FS.AST= 000000	LF.PAC= 000200	N\$SMCP= 000001
CE.ERR= 100370	CS.TMO= 020000	FS.CIB= 002000	LF.RDY= 040000	N\$SMLL= 000001
CE.ILN= 100350	CS.XUR= 000004	FS.CRA= 001000	LF.REA= 010000	N\$SMOV= 000010
CE.LTO= 100356	C\$CKP= 000000	FS.DIS= 013000	LF.SER= 000040	N\$SNCT= 000001
CE.MOP= 100372	C\$ORE= 000400	FS.DVC= 001000	LF.TIM= 000010	N\$SPEM= 000001
CE.NTE= 100361	C\$RSH= 177564	FS.ENB= 012000	LF.UNL= 020000	P\$P45= 000000
CE.RTE= 100376	C.ADD 000034	FS.EXI= 001000	LF.X2P= 000000	P\$SWRD= 000000
CE.SRC= 100364	C.BID 000003	FS.GET= 006000	LN.CLO= 000000	Q\$SOPT= 000010
CE.STP= 100352	C.BUF 000014	FS.HLT= 000000	LN.DUM= 000005	RDBRT = ***** GX
CE.TME= 100334	C.BUF1 000014	FS.INI= 000000	LN.LOA= 000004	R\$DER= 000000
CE.TMO= 100374	C.BUF2 000024	FS.KIL= 000000	LN.LOO= 000033	R\$K11= 000001
CE.UNS= 100344	C.CNT 000020	FS.LCL= 100000	LN.OAU= 000003	R\$SND= 000000
CF.CHN= 000001	C.CNT1 000020	FS.LTM= 001000	LN.OFF= 000001	R\$11M= 000000
CF.EDM= 000004	C.CNT2 000030	FS.MNY= 004000	LN.ON = 000000	SF.ACT= 000200
CF.HDR= 000020	C.FLG 000022	FS.MSN= 014000	LN.OOP= 000004	SF.ENA= 000100
CF.LB = 100000	C.FLG1 000022	FS.REA= 001000	LN.OPE= 000001	SF.LPB= 000004
CF.LIN= 000002	C.FLG2 000032	FS.RET= 000000	LN.REF= 000002	SF.MFL= 000040
CF.SDM= 000010	C.FNC 000010	FS.REZ= 003000	LN.SER= 000002	SF.PAC= 000020
CF.SYN= 000040	C.LIN 000006	FS.RLB= 002000	LN.STA= 000017	SF.REA= 000010
CF.TRN= 000100	C.LNK 000000	FS.RNG= 011000	LN.SUB= 000360	SF.SER= 000001
CM.CIR= 000002	C.MOD 000011	FS.RST= 000000	LN.TRI= 000006	SF.SVC= 000002
CM.FMT= 100000	C.NSP 000004	FS.RTN= 001000	L\$ASG= 000000	SF.UNL= 000040
CM.HRD= 000002	C.PRO 000042	FS.SET= 005000	L\$DRV= 000000	SLTMA = ***** GX
CM.LIN= 000000	C.RSV 000002	FS.SFC= 005000	L\$P11= 000001	SRSTD = ***** GX
CM.LOO= 000001	C.STA 000007	FS.SFR= 006000	L\$11R= 000000	S\$WRG= 000000
CM.XLO= 000004	C.STS 000012	FS.SFS= 004000	L.COST 000015	S\$YSZ= 007600
CP.DCF= 000040	C.URM 177776	FS.SPW= 040000	L.CTL 000012	S.COST 000001
CP.HDL= 000007	C.XACP 000004	FS.STM= 000000	L.CVA 177776	S.FLG 000000
CP.PS = 177400	C.XID 000035	FS.STP= 002000	L.DDM 000002	S.LEN 000004
CP.PSI= 000200	C.XLEN 000044	FS.STR= 001000	L.DDS 000004	S.NMST 000002
CP.XCF= 000100	C.XPLI 000040	FS.TRM= 003000	L.DLC 000003	S.OWNR 000003
CP.2FR= 000030	C.XPT 000034	FS.WLB= 001000	L.DLM 000006	T\$KMG= 000000
CS.ABO= 000100	C.XSVC 000042	FS.XKL= 002000	L.DLS 000010	T\$MIN= 000000
CS.BRO= 000002	C.XTC 000037	FS.XOF= 010000	L.FLG 000000	V\$CTR= 001000
CS.BUF= 000200	C.X25 000036	FS.XON= 007000	L.KRBA 000016	X\$DBT= 000000
CS.CES= 000002	DUMMY 000026R	FS.ZER= 002000	L.LEN = 000022	X\$MPC= 000001
CS.CHN= 000010	D\$BUG= 177514	F\$LVL= 000001	L.MPF 000022	\$AUXTB 000000RG
CS.CMP= 000200	D\$ISK= 000000	G\$TPP= 000000	L.NMST 000020	\$BFTN= ***** GX
CS.DCR= 000400	D\$L11= 000001	G\$TSS= 000000	L.NSTA 000014	\$.\$\$\$= 000034
CS.DEF= 000004	D\$SYNC= 000000	G\$TTK= 000000		

```

193      .SBTTL POWERFAIL RECOVERY DISPATCH TO DDM MODULES
194
195      :+
196      :*-PWRFL-POWERFAIL RECOVERY DISPATCH TO DDM MODULES
197
198      :
199      :
200      :
201      :
202      :
203      :
204      :
205      :
206      :
207      :
208      :
209      :
210      :
211      :
212      :
213      :
214      :
215      :
216      :
217      :
218      :
219      :
220      :
221      :
222      :
223      :
224      :
225      :
226      :
227      :
228      :
229      :
230      :
231      :
232      :
233      :
234      :
235      :
236      :
237      :
238      :
239      :
240      :
241      :
242      :
243      :
244      :
245      :
246      :
247      :
248      :
249      :

```

THIS ROUTINE IS INVOKED FROM THE DRIVER POWERFAIL ENTRY POINTS TO DISPATCH TO DDM MODULES TO PERFORM POWERFAIL RECOVERY. IF THE KMC MICROCODE LOADER TASK IS INSTALLED, IT WILL BE REQUESTED TO RUN.

```

PWRFL: MOV    #~1,@PWRFL1      ; STOP TIMERS RUNNING TO DDM PROCESSES
        MOV    @SLTNM,-(SP)    ; SET UP COUNT OF LINES TO SCAN

10$:    MOV    (SP),R3          ; GET NEXT SYSTEM LINE #
        DEC    R3              ;
        ASL    R3              ; FORM WORD OFFSET
        ADD    @SLTMA,R3       ; POINT INTO SYSTEM LINE INDEX TABLE
        MOV    (R3),R3         ; AND GET ADDRESS OF SYSTEM LINE TABLE
        TST    (R3)            ; IS THE LINE ACTIVE?
        BPL    20$             ; IF PL, NO

        .IF DF R$$MPL
        .IF NDF R$$PRO

        BIT    #F2.MP,@FMSK2   ; IS THIS A MULTIPROCESSOR?
        BEQ    15$             ; BR IF NO
        MOV    L.KRBA(R3),R5    ; GET ADDRESS OF KRB
        MOV    @CPURM,-(SP)     ; GET ADDRESS OF CPU URM TABLE
        BIT    K.URM(R5),@ (SP)+ ; ARE WE RUNNING ON THE CORRECT PROCESSOR?
        BEQ    20$             ; IF EQ, NO

15$:    .ENDC
        .ENDC

        MOV    L.DDS(R3),R5     ; GET LINE TABLE ADDRESS
        MOVB   L.DDM(R3),R2     ; AND DDM PDV INDEX
        MOV    DDFNC,R3         ; POINT TO POWERFAIL FUNCTION CODE
        CALL   @PDSPL           ; DISPATCH TO DDM

20$:    DEC    (SP)             ; REDUCE COUNT OF LINES TO SCAN
        BNE    10$             ; LOOP IF MORE TO GO
        TST    (SP)+           ; CLEAN UP THE STACK

        .IF NDF I$$AS

        MOV    #KMCL,R3        ; POINT TO KMC LOADER TASK NAME
        CALL   @SRSTD          ; SCAN TO FIND TASK'S TCB
        BCS    30$             ; IF CS, TASK NOT INSTALLED
        CLR    R1              ; CLEAR DEFAULT UIC
        CALL   @TSKRT          ; REQUEST TASK TO RUN

        .ENDC

30$:    MOV    @SLTNM,@PWRFL1   ; START DISCONNECT NOTIFICATION ON ALL LINES
        RETURN

        .IF NDF I$$AS
        .RAD50 /KMCL../       ; TASK NAME OF KMC MICROCODE LOADER

```

```

256 .SBTTL $MVFBF - MOVE FROM UNMAPPED BUFFER TO A MAPPED BUFFER
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308

```

**-\$MVFBF - MOVE FROM UNMAPPED BUFFER TO A MAPPED BUFFER
 THIS ROUTINE IS CALLED TO MOVE A BLOCK OF MEMORY (LESS THAN
 4K WORDS) FROM AN UNMAPPED BUFFER TO A BUFFER THAT IS CURRENTLY
 MAPPED (EITHER IN SYSTEM DYNAMIC SPACE OR IN THE COMM BUFFER POOL).
 THE ROUTINE UNMAPS THE REQUESTING PROCESS AND USES ARPS TO MAP
 TO THE UNMAPPED "FROM" BUFFER.
 CALLING FORMAT:
 JSR R1,\$MVFBF
 INPUTS:
 R2 = VIRTUAL ADDRESS OF MAPPED "TO" BUFFER
 R3 = NUMBER OF BYTES TO MOVE
 ON STACK:

R1	ORIGINAL CONTENTS OF R1 BEFORE CALL
VA	16-BIT VIRTUAL ADDRESS OF UNMAPPED "FROM" BUFFER
BIAS	RELOCATION BIAS OF UNMAPPED "FROM" BUFFER (MAPPED SYSTEMS ONLY)

 OUTPUTS:
 R2 = UPDATED ADDRESS OF "TO" BUFFER
 POINTS TO LAST BYTE MOVED +1
 R3 = ZERO
 STILL MAPPED TO "TO" BUFFER VIA KISAR6
 REGISTERS MODIFIED:
 R2 & R3
 -
 \$MVFBF::

000216	016746	000000G	MOV	KISAR5,-(SP)	: SAVE CURRENT PROCESS MAPPING
000216	016667	000006	MOV	6(SP),KISAR5	: MAP TO "FROM" BUFFER
000222	010166	000006	MOV	R1,6(SP)	: SAVE RETURN ADDRESS
000230	016601	000004	MOV	4(SP),R1	: GET "FROM" BUFFER VIRTUAL ADDRESS
000234	022701	140000	CMP	#140000,R1	: IS THE "FROM" BUFFER IN THE EXEC POOL?
000240	101002		BHI	15\$: IF H1, YES THEN DON'T ALTER THE VIRTUAL ADDRESS
000244	162701	020000	SUB	#20000,R1	: SET VA FOR BIAS VIA KISAR5
000252	112122		15\$:	MOVB	(R1)+(R2)+
000254				SUB	R3,15\$
000260	012667	000000G	20\$:	MOV	(SP)+,KISAR5
000264	012601			MOV	(SP)+,R1
000266	005726			TST	(SP)+
000270				RETURN	: CLEAN UP STACK
					: RETURN

 .DSABL LSB

FILEID**CETIM

K 2

CCCCCCCC	EEEEEEEEEE	TTTTTTTTTT	IIIIII	MM	MM	
CCCCCCCC	EEEEEEEEEE	TTTTTTTTTT	IIIIII	MM	MM	
CC	EE	TT	II	MMMM	MMMM	
CC	EE	TT	II	MMMM	MMMM	
CC	EE	TT	II	MM	MM	MM
CC	EE	TT	II	MM	MM	MM
CC	EEEEEEEE	TT	II	MM	MM	
CC	EEEEEEEE	TT	II	MM	MM	
CC	EE	TT	II	MM	MM	
CC	EE	TT	II	MM	MM	
CC	EE	TT	II	MM	MM
CC	EE	TT	II	MM	MM
CCCCCCCC	EEEEEEEEEE	TT	IIIIII	MM	MM
CCCCCCLC	EEEEEEEEEE	TT	IIIIII	MM	MM

LL	SSSSSSSS	TTTTTTTTTT
LL	SSSSSSSS	TTTTTTTTTT
LL	SS	TT
LL	SS	TT
LL	SS	TT
LL	SS	TT
LL	SSSSSS	TT
LL	SSSSSS	TT
LL	SS	TT
LL	SS	TT
LL	SS	TT
LL	SS	TT
LLLLLLLLLL	S' SSSSSS	TT
LLLLLLLLLL	SSSSSSSS	TT

```
58 000000      OPTDF$          ; DEFINE CEX OPTIONS FLAGS
59
60 000000      .PSECT CEXCOM
61
62             .ENDC  ; NDF R$$MPL
63
64             .MCALL SYNDF$      ; DEFINE SYNCH AND PAD SYMBOLS
65
66 000000      SYNDF$ <:;>,<=>
```

```
1      .IF DF K$$G11
2      .TITLE STCRCK
3      .IFF
4      .IF DF F$$AST
5      .TITLE STCRCF
6      .IFF
7      .TITLE STCRC
8      .ENDC
9      .ENDC
10     .IDENT /V05.00/
```

```
11
12     +
13     COPYRIGHT (C) 1978,1979,1980, 1982, 1983, 1985 BY
14     DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
```

```
15     THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
16     SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
17     INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR
18     ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE
19     MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH
20     SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE
21     TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN
22     IN DEC.
```

```
23     THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
24     NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
25     EQUIPMENT CORPORATION.
```

```
26     DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
27     ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
```

```
28     MODULE DESCRIPTION
```

```
29     CRC-16 CALCULATION ROUTINES
```

```
30     DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
```

```
31     IDENT HISTORY:
```

- ```
32 1.00 10-FEB-78
33 VERSION 2.0 RELEASE
34
35 2.00 14-DEC-79
36 DECNET-11M/S V3.0
37 DECNET-11M-PLUS V1.0
38
39 3.00 16-APR-82
40 DECNET-11M V3.1
41 DECNET-11M-PLUS V1.1
42
43 4.00 07-NOV-83
44 DECNET-11M V4.0
45 DECNET-11M-PLUS V2.0
46
47 5.00 22-JUL-85
48 DECnet-11M/S V4.2
49 DECnet-11M-Plus V3.0
50
51
52
53
54
55
56
57
```



|     |       |        |
|-----|-------|--------|
| 488 | .WORD | 152301 |
| 489 | .WORD | 152601 |
| 490 | .WORD | 12500  |
| 491 | .WORD | 153401 |
| 492 | .WORD | 13700  |
| 493 | .WORD | 13200  |
| 494 | .WORD | 153101 |
| 495 | .WORD | 151001 |
| 496 | .WORD | 11300  |
| 497 | .WORD | 11600  |
| 498 | .WORD | 151501 |
| 499 | .WORD | 10400  |
| 500 | .WORD | 150701 |
| 501 | .WORD | 150201 |
| 502 | .WORD | 10100  |
| 503 | .WORD | 170001 |
| 504 | .WORD | 30300  |
| 505 | .WORD | 30600  |
| 506 | .WORD | 170501 |
| 507 | .WORD | 31400  |
| 508 | .WORD | 171701 |
| 509 | .WORD | 171201 |
| 510 | .WORD | 31100  |
| 511 | .WORD | 33000  |
| 512 | .WORD | 173301 |
| 513 | .WORD | 173601 |
| 514 | .WORD | 33500  |
| 515 | .WORD | 172401 |
| 516 | .WORD | 32700  |
| 517 | .WORD | 32200  |
| 518 | .WORD | 172101 |
| 519 | .WORD | 36000  |
| 520 | .WORD | 176301 |
| 521 | .WORD | 176601 |
| 522 | .WORD | 36500  |
| 523 | .WORD | 177401 |
| 524 | .WORD | 37700  |
| 525 | .WORD | 37200  |
| 526 | .WORD | 177101 |
| 527 | .WORD | 175001 |
| 528 | .WORD | 35300  |
| 529 | .WORD | 35600  |
| 530 | .WORD | 175501 |
| 531 | .WORD | 34400  |
| 532 | .WORD | 174701 |
| 533 | .WORD | 174201 |
| 534 | .WORD | 34100  |
| 535 | .WORD | 24000  |
| 536 | .WORD | 164301 |
| 537 | .WORD | 164601 |
| 538 | .WORD | 24500  |
| 539 | .WORD | 165401 |
| 540 | .WORD | 25700  |
| 541 | .WORD | 25200  |
| 542 | .WORD | 165101 |
| 543 | .WORD | 167001 |
| 544 | .WORD | 27300  |

|    |   |                       |
|----|---|-----------------------|
| 58 | : | DECnet-Micro/RSX V1.0 |
| 59 | : |                       |
| 60 | : |                       |
| 61 | : |                       |

|     |        |        |       |        |
|-----|--------|--------|-------|--------|
| 545 | 000560 | 027600 | .WORD | 27600  |
| 546 | 000562 | 167501 | .WORD | 167501 |
| 547 | 000564 | 026400 | .WORD | 26400  |
| 548 | 000566 | 166701 | .WORD | 166701 |
| 549 | 000570 | 166201 | .WORD | 166201 |
| 550 | 000572 | 026100 | .WORD | 26100  |
| 551 | 000574 | 162001 | .WORD | 162001 |
| 552 | 000576 | 022300 | .WORD | 22300  |
| 553 | 000600 | 022600 | .WORD | 22600  |
| 554 | 000602 | 162501 | .WORD | 162501 |
| 555 | 000604 | 023400 | .WORD | 23400  |
| 556 | 000606 | 163701 | .WORD | 163701 |
| 557 | 000610 | 163201 | .WORD | 163201 |
| 558 | 000612 | 023100 | .WORD | 23100  |
| 559 | 000614 | 021000 | .WORD | 21000  |
| 560 | 000616 | 161301 | .WORD | 161301 |
| 561 | 000620 | 161601 | .WORD | 161601 |
| 562 | 000622 | 021500 | .WORD | 21500  |
| 563 | 000624 | 160401 | .WORD | 160401 |
| 564 | 000626 | 020700 | .WORD | 20700  |
| 565 | 000630 | 020200 | .WORD | 20200  |
| 566 | 000632 | 160101 | .WORD | 160101 |
| 567 | 000634 | 120001 | .WORD | 120001 |
| 568 | 000636 | 060300 | .WORD | 60300  |
| 569 | 000640 | 060600 | .WORD | 60600  |
| 570 | 000642 | 120501 | .WORD | 120501 |
| 571 | 000644 | 061400 | .WORD | 61400  |
| 572 | 000646 | 121701 | .WORD | 121701 |
| 573 | 000650 | 121201 | .WORD | 121201 |
| 574 | 000652 | 061100 | .WORD | 61100  |
| 575 | 000654 | 063000 | .WORD | 63000  |
| 576 | 000656 | 123301 | .WORD | 123301 |
| 577 | 000660 | 123601 | .WORD | 123601 |
| 578 | 000662 | 063500 | .WORD | 63500  |
| 579 | 000664 | 122401 | .WORD | 122401 |
| 580 | 000666 | 062700 | .WORD | 62700  |
| 581 | 000670 | 062200 | .WORD | 62200  |
| 582 | 000672 | 122101 | .WORD | 122101 |
| 583 | 000674 | 066000 | .WORD | 66000  |
| 584 | 000676 | 126301 | .WORD | 126301 |
| 585 | 000700 | 126601 | .WORD | 126601 |
| 586 | 000702 | 066500 | .WORD | 66500  |
| 587 | 000704 | 127401 | .WORD | 127401 |
| 588 | 000706 | 067700 | .WORD | 67700  |
| 589 | 000710 | 067200 | .WORD | 67200  |
| 590 | 000712 | 127101 | .WORD | 127101 |
| 591 | 000714 | 125001 | .WORD | 125001 |
| 592 | 000716 | 065300 | .WORD | 65300  |
| 593 | 000720 | 065600 | .WORD | 65600  |
| 594 | 000722 | 125501 | .WORD | 125501 |
| 595 | 000724 | 064400 | .WORD | 64400  |
| 596 | 000726 | 124701 | .WORD | 124701 |
| 597 | 000730 | 124201 | .WORD | 124201 |
| 598 | 000732 | 064100 | .WORD | 64100  |
| 599 | 000734 | 074000 | .WORD | 74000  |
| 600 | 000736 | 134301 | .WORD | 134301 |
| 601 | 000740 | 134601 | .WORD | 134601 |

```

63
64
65
66
67 000000
68 000000
69
70
71
72
73
74
75 000000
76
77
78
79
80
81
82
83
84
85
86
87
88

 .SBTTL MACRO DEFINITIONS
 .MCALL CCBDF$,NHWDF$,SAVRG,RESRG
 CCBDF$; DEFINE CCB OFFSETS
 NHWDF$; DEFINE HARDWARE REGISTERS

 ;
 ; DEFINE LOCAL MACROS
 ;
 .IF DF P$$40!P$$34!P$$45!P$$70!L$$S11!M$$MGE
 R$$E1S=0 ; PROCESSOR HAS XOR INSTRUCTION
 .ENDC
 .IF NDF R$$E1S
 .MACRO XOR,A,B
 MOV A,R3 ; COPY A
 BIC B,R3 ; A <- A AND NOT B
 BIC A,B ; B <- B AND NOT A
 BIS R3,B ; B <- A OR B
 .ENDM XOR
 .ENDC

```

|     |       |        |
|-----|-------|--------|
| 602 | .WORD | 74500  |
| 603 | .WORD | 135401 |
| 604 | .WORD | 75700  |
| 605 | .WORD | 75200  |
| 606 | .WORD | 135101 |
| 607 | .WORD | 137001 |
| 608 | .WORD | 77300  |
| 609 | .WORD | 77600  |
| 610 | .WORD | 137501 |
| 611 | .WORD | 76400  |
| 612 | .WORD | 136701 |
| 613 | .WORD | 136201 |
| 614 | .WORD | 76100  |
| 615 | .WORD | 132001 |
| 616 | .WORD | 72300  |
| 617 | .WORD | 72600  |
| 618 | .WORD | 132501 |
| 619 | .WORD | 73400  |
| 620 | .WORD | 133701 |
| 621 | .WORD | 133201 |
| 622 | .WORD | 73100  |
| 623 | .WORD | 71000  |
| 624 | .WORD | 131301 |
| 625 | .WORD | 131601 |
| 626 | .WORD | 71500  |
| 627 | .WORD | 130401 |
| 628 | .WORD | 70700  |
| 629 | .WORD | 70200  |
| 630 | .WORD | 130101 |
| 631 | .WORD | 50000  |
| 632 | .WORD | 110301 |
| 633 | .WORD | 110601 |
| 634 | .WORD | 50500  |
| 635 | .WORD | 111401 |
| 636 | .WORD | 51700  |
| 637 | .WORD | 51200  |
| 638 | .WORD | 111101 |
| 639 | .WORD | 113001 |
| 640 | .WORD | 53300  |
| 641 | .WORD | 53600  |
| 642 | .WORD | 113501 |
| 643 | .WORD | 52400  |
| 644 | .WORD | 112701 |
| 645 | .WORD | 112201 |
| 646 | .WORD | 52100  |
| 647 | .WORD | 116001 |
| 648 | .WORD | 56300  |
| 649 | .WORD | 56600  |
| 650 | .WORD | 116501 |
| 651 | .WORD | 57400  |
| 652 | .WORD | 117701 |
| 653 | .WORD | 117201 |
| 654 | .WORD | 57100  |
| 655 | .WORD | 55000  |
| 656 | .WORD | 115301 |
| 657 | .WORD | 115601 |
| 658 | .WORD | 55500  |

```

63 +
64 ***$BFRTN-SYSTEM LEVEL ROUTINE TO SATISFY A BUFFER WAIT REQUEST
65
66 INPUTS:
67
68 R3 = SUB-FUNCTION CODE
69 R4 = ADDRESS OF BUFFER TO SATISFY A WAIT REQUEST
70
71 SCAN FOR NEXT SYSTEM LINE WHICH HAS AN OUTSTANDING BUFFER WAIT REQUEST
72 AND GIVE THE BUFFER TO THE DEVICE DRIVER MODULE WITH A RECEIVE ENABLE
73 FUNCTION CODE.
74
75 -
76
77 000000 $BFRTN:: ; REF LABEL
78 .IF DF R$$MPL
79 .IF NDF R$$PRO
80
81 BIT #F2.MP,@FMSK2 ; IS THIS A MULTIPROCESSOR?
82 BEQ 5$; BR IF NO
83 TSTB R3 ; IS THIS BUFFER TO GO TO THE DEVICE?
84 BEQ MPBRTN ; IF EQ, YES
85
86 CALL @MPSAV ; BYPASS THE CACHE
87
88 5$:
89 .ENDC
90 .ENDC
91 000000 012764 100000 000022 MOV #CF.LB,C.FLG(R4); RESET CCB ATTRIBUTES FOR BUFFER
92 000006 017764 000000G 000020 MOV @RDBSZ,C.CNT(R4); ...
93 000014 017764 000000G 000030 MOV @RDBSZ,C.CNT2(R4)
94 000022 112764 000004 000003 MOVB #CB.RDB,C.BID(R4)
95
96 000030 042777 100000 000000G 10$: MTPS #PR7 ; INHIBIT INTERRUPTS
97 000036 001453 BIC #100000,@RDQCT ; CLEAR BUFFER QUEUED FLAG AND TEST FOR WAITERS
98 000044 BEQ 30$; IF EQ NO, RETURN BUFFER TO THE POOL
99
100 .IF DF R$$11D!I$$AS
101
102 MTPS #PR3
103
104 .IFF
105
106 000046 MTPS #PRO ; ENABLE INTERRUPTS
107
108 .ENDC
109
110 000054 017702 000000G MOV @RDQSL,R2 ; GET NEXT SYSTEM LINE NUMBER
111 000060 001003 BNE 20$; IF NE NUMBER IS STILL VALID
112 000062 017777 000000G 000000G MOV @SLTNM,@RDQSL ; RESET HIGHEST SYSTEM LINE NUMBER PLUS ONE
113 000070 005377 000000G 20$: DEC @RDQSL ; COMPUTE NEXT SYSTEM LINE NUMBER
114 000074 010203 MOV R2,R3 ; COPY SYSTEM LINE NUMBER
115
116 .IF DF N$$1LN
117
118 MOV @SLTMA,R2 ; GET ADDRESS OF SLT ENTRY
119 MOV (R2),R2 ; ...

```

```
156 MPLCK:: .WORD $MPLCK
157 MPSAV:: .WORD $MPSAV
158 .ENDC ; NDF R$$PRO
159
160 PROC2:: .WORD $PROC2
161 QFORK:: .WORD $QFORK
162 .ENDC
163
164 000070 $AUXVL==.-1-$AUXVT/2
165 000001 .END
```

```

193 .SBTTL POWERFAIL RECOVERY DISPATCH TO DDM MODULES
194 ;+
195 ;**--PWRFL-POWERFAIL RECOVERY DISPATCH TO DDM MODULES
196 ;
197 ; THIS ROUTINE IS INVOKED FROM THE DRIVER POWERFAIL ENTRY POINTS TO
198 ; DISPATCH TO DDM MODULES TO PERFORM POWERFAIL RECOVERY. IF THE KMC
199 ; MICROCODE LOADER TASK IS INSTALLED, IT WILL BE REQUESTED TO RUN.
200 ; -
201 PWRFL: MOV #-1,@PWRFL1 ; STOP TIMERS RUNNING TO DDM PROCESSES
202 MOV @SLTNM,-(SP) ; SET UP COUNT OF LINES TO SCAN
203
204 10$: MOV (SP),R3 ; GET NEXT SYSTEM LINE #
205 DEC R3 ;
206 ASL R3 ; FORM WORD OFFSET
207 ADD @SLTMA,R3 ; POINT INTO SYSTEM LINE INDEX TABLE
208 MOV (R3),R5 ; AND GET ADDRESS OF SYSTEM LINE TABLE
209 TST (R3) ; IS THE LINE ACTIVE?
210 BPL 20$; IF PL, NO
211
212 .IF DF R$$MPL
213 .IF NDF R$$PRO
214
215 BIT #F2.MP,@FMSK2 ; IS THIS A MULTIPROCESSOR?
216 BEQ 15$; BR IF NO
217 MOV L.KRBA(R3),R5 ; GET ADDRESS OF KRB
218 MOV @CPURM,-(SP) ; GET ADDRESS OF CPU URM TABLE
219 BIT K.URM(R5),@ (SP)+ ; ARE WE RUNNING ON THE CORRECT PROCESSOR?
220 BEQ 20$; IF EQ, NO
221
222 15$: .ENDC
223 .ENDC
224
225 MOV L.DDS(R3),R5 ; GET LINE TABLE ADDRESS
226 MOV L.DDM(R3),R2 ; AND DDM PDV INDEX
227 MOV DDFNC,R3 ; POINT TO POWERFAIL FUNCTION CODE
228 CALL @PDSPL ; DISPATCH TO DDM
229
230 20$: DEC (SP) ; REDUCE COUNT OF LINES TO SCAN
231 BNE 10$; LOOP IF MORE TO GO
232 TST (SP)+ ; CLEAN UP THE STACK
233
234 .IF NDF I$$AS
235
236 MOV #KMCL,R3 ; POINT TO KMC LOADER TASK NAME
237 CALL @SRSTD ; SCAN TO FIND TASK'S TCB
238 BCS 30$; IF CS, TASK NOT INSTALLED
239 CLR R1 ; CLEAR DEFAULT UIC
240 CALL @TSKRT ; REQUEST TASK TO RUN
241
242 .ENDC
243
244 30$: MOV @SLTNM,@PWRFL1 ; START DISCONNECT NOTIFICATION ON ALL LINES
245 RETURN
246
247 KMCL: .IF NDF I$$AS
248 .RAD50 /KMCL../ ; TASK NAME OF KMC MICROCODE LOADER
249

```



75  
76  
77 000000  
78 000000

.MCALL SLTDF\$,CCBDF\$,SAVRG,RESRG

SLTDF\$  
CCBDF\$

; DEFINE SYSTEM LINE TABLE OFFSETS  
; DEFINE CCB OFFSETS

\*\*FILE\*\*ID\*\*AXDSPM

|        |    |    |          |          |          |    |    |      |
|--------|----|----|----------|----------|----------|----|----|------|
| AAAAAA | XX | XX | DDDDDDDD | SSSSSSSS | PPPPPPPP | MM | MM |      |
| AAAAAA | XX | XX | DDDDDDDD | SSSSSSSS | PPPPPPPP | MM | MM |      |
| AA     | AA | XX | DD       | DD       | PP       | MM | MM |      |
| AA     | AA | XX | DD       | DD       | PP       | MM | MM |      |
| AA     | AA | XX | DD       | DD       | PP       | MM | MM |      |
| AA     | AA | XX | DD       | DD       | PP       | MM | MM |      |
| AA     | AA | XX | DD       | DD       | PP       | MM | MM |      |
| AAAAAA | XX | XX | DD       | DD       | PP       | MM | MM |      |
| AAAAAA | XX | XX | DD       | DD       | PP       | MM | MM |      |
| AA     | AA | XX | DD       | DD       | PP       | MM | MM | .... |
| AA     | AA | XX | DD       | DD       | PP       | MM | MM | .... |
| AA     | AA | XX | DDDDDDDD | SSSSSSSS | PP       | MM | MM | .... |
| AA     | AA | XX | DDDDDDDD | SSSSSSSS | PP       | MM | MM | .... |

|            |          |           |
|------------|----------|-----------|
| LL         | SSSSSSSS | TTTTTTTTT |
| LL         | SSSSSSSS | TTTTTTTTT |
| LL         | SS       | TT        |
| LL         | SS       | TT        |
| LL         | SS       | TT        |
| LL         | SS       | TT        |
| LL         | SSSSSS   | TT        |
| LL         | SSSSSS   | TT        |
| LL         | SS       | TT        |
| LL         | SS       | TT        |
| LL         | SS       | TT        |
| LL         | SS       | TT        |
| LL         | SS       | TT        |
| LLLLLLLLLL | SSSSSSSS | TT        |
| LLLLLLLLLL | SSSSSSSS | TT        |

AXDSPM - AUXILLIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 10:59 <sup>K 15</sup> Page 10-3  
Symbol table

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000244 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 16127 Words ( 63 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:43.46  
DB2:AXDSPM.T47,[131,134]AXDSPM/CR/-SP=DB2:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]T47,AXDSP

AXDSPP - AUXILLIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 K 16  
POWERFAIL RECOVERY DISPATCH TO DDM MODULES

250  
251  
252

.ENDC

.ENDC

```

310 .SBTTL $CALLX - MAPPED SUBROUTINE CALL
311
312 ;+
313 ;**-$CALLX-MAPPED SUBROUTINE CALL TO ANOTHER PROCESS
314
315 CALLING SEQUENCE:
316 JSR R5,$CALLX
317 .WORD <ADDRESS OF SUBROUTINE TO CALL>
318 .RAD50 <PROCESS NAME>
319
320 ; NOTE THAT THIS SUBROUTINE ASSUMES THAT THE NAMED PROCESS EXISTS IN THE
321 ; SYSTEM AND DOES NOT CHECK IF THE PROCESS IS LOADED.
322 ; -
323
324 000006 .IIF NDF K$$DAS OFS=6 ; STACK OFFSET FOR R5
325 .IIF DF K$$DAS OFS=10 ; STACK OFFSET FOR R5
326
327 000272 016746 000000G $CALLX::MOV $CMPDV,-(SP) ; SAVE CURRENT PDV INDEX
328 000276 016746 000000G MOV KISAR5,-(SP) ; AND CURRENT PROCESS MAPPING
329
330 .IF DF K$$DAS
331 MOV KINAR5,-(SP) ; SAVE INSTRUCTION MAPPING
332 .ENDC ; DF K$$DAS
333
334 000302 012543 MOV (R5)+,-(SP) ; SAVE DESTINATION SUBROUTINE ADDRESS
335 000304 SAVRG <R1,R2> ; GET 2 FREE REGISTERS
336 000310 016701 000000G MOV $PDVTA,R1 ; POINT TO PDV INDEX TABLE
337 000314 012102 MOV (R1)+,R2 ; GET POINTER TO NEXT PDV
338 000316 001776 BEQ 10$; IGNORE ZERO ENTRIES
339 000320 026215 000004 CMP Z,NAM(R2),(R5) ; DO WE HAVE A MATCH?
340 000324 001373 BNE 10$; IF NE, NO
341 000326 005725 TST (R5)+ ; SKIP OVER PROCESS NAME
342 000330 011267 000000G MOV (R2),KISAR5 ; MAP TO DESTINATION PROCESS
343
344 .IF DF K$$DAS
345 MOV (R2),KINAR5 ; ALSO MAP INSTRUCTION SPACE
346 .ENDC ; DF K$$DAS
347
348 000334 005741 TST -(R1) ; BACK UP PDV INDEX TABLE POINTER
349 000336 166701 000000G SUB $PDVTA,R1 ; COMPUTE PDV INDEX
350 000342 010167 000000G MOV R1,$CMPDV ; AND SAVE IT
351 000346 016602 000012 MOV OFS+4(SP),R2 ; SWAP OLD R5 AND RETURN ADDRESS
352 000352 010366 000012 MOV R5,OF5+4(SP) ; ...
353 000356 010205 MOV R2,R5 ; RESTORE REGISTERS
354 000360 RESRG <R2,R1>
355
356 000364 CALL @(<SP>)+ ; CALL THE SUBROUTINE
357
358 .IF DF K$$DAS
359 MOV (SP)+,KINAR5 ; RESTORE INSTRUCTION MAPPING
360 .ENDC ; DF K$$DAS
361
362 000366 012667 000000G MOV (SP)+,KISAR5 ; RESTORE PROCESS MAPPING
363 000372 012667 000000G MOV (SP)+,$CMPDV ; RESTORE CURRENT PDV INDEX
364 000376 RETURN

```

CETIM MACRO V05.03b Friday 28-Jun-85 18:21  
Table of contents

L 2

5- 61 TIMEOUT PROCESSOR  
6- 104 DISPATCH A PROCESS TIMEOUT

```

68 ;+
69 ; COMMUNICATION SYSTEM COMMON DATA BASE
70 ; (ORDERING IMPACTS VNP)
71 ; -
72
73
74 000000 $PDVTA::BLKW 1 ; ADDRESS OF PDV ADDRESS TABLE
75 000002 $SLTMA::BLKW 1 ; ADDRESS OF SLT ADDRESS TABLE
76 000004 $LLCTA::BLKW 1 ; ADDRESS OF LLC REVERSE MAPPING TABLE
77 000006 $PDVNM::BLKW 1 ; NUMBER OF PDV ENTRIES
78 000010 $SLTNM::BLKW 1 ; NUMBER OF SYSTEM LINES
79 000012 $CCBNM::BLKW 1 ; # OF CCB'S ALLOCATED
80 000014 $CCBSZ::BLKW 1 ; # BYTES / CCB
81 000016 $RDBNM::BLKW 1 ; # OF RDB'S ALLOCATED
82 000020 $RDBSZ::BLKW 1 ; # BYTES / RDB
83 000022 $SDBNM::BLKW 1 ; # OF SDB'S ALLOCATED
84 000024 $SDBSZ::BLKW 1 ; # BYTES / SDB
85 000026 $CCBCT::BLKW 1 ; COUNTER OF CURRENT # OF CCB'S IN POOL
86 000030 $CCBAF::BLKW 1 ; COUNTER OF # OF CCB ALLOCATION FAILURES
87 000032 $LDBAF::BLKW 1 ; COUNTER OF # OF LDB ALLOCATION FAILURES
88 000034 $CCBAL::BLKW 1 ; COUNTER OF # OF DYNAMIC CCB ALLOCATIONS
89 000036 $RDBTH::BLKW 1 ; POOL THRESHOLD FOR LDB ALLOCATION FAILURE
90 000040 $CMPDV::BLKW 1 ; PDV INDEX OF CURRENT DISPATCHED PROCESS
91 000042 $ZTIME::BLKW 1 ; TIME SINCE COUNTERS LAST ZEROED
92 000044 $ZTIM2::BLKW 1 ; TIME SINCE SYSTEM LAST ZEROED
93
94 .IF DF M$$PRO
95 .WORD 0 ; UNIBUS RUN MASK
96 .ENDC
97
98 000046 000000 $CMFRK::WORD 0 ; INITIAL LINK MARKS FORK BLOCK FREE
99 000050 000000 .WORD 0 ; ADDRESS OF FORK PROCESS
100 000052 000000 .WORD 0 ; FORK PROCESS QUEUE
101 000054 000052 .WORD -2
102
103 000056 000000 .WORD 0 ; APR5 BIAS FOR FORK LEVEL DISPATCH
104 ; FREE WORD ON UNMAPPED SYSTEMS
105
106 000060 $T50Q:: ; PHASE III COMPATIBILITY
107 000060 000000 $T100C::WORD 0 ; 100 MSEC TIMER QUEUE
108 000062 000060 .WORD -2
109
110 000064 $STMFC::
111 .IF NDF R$$MPL
112 000064 000010 .WORD FC.TIM+FS.STM ; FUNCTION CODE TO DISPATCH 50 MSEC TIMEOUTS
113 .IFF
114 .WORD 0
115 .ENDC ; R$$MPL
116
117 000066 $T50CL:: ; PHASE III COMPATIBILITY
118 000066 000000 000000 000000 $T100C::WORD 0,0,0,0,0 ; 100 MSEC CLOCK QUEUE ENTRY
119 000074 000000 000000 .WORD 0 ; ADDRESS OF PROCESSING ROUTINE
120 000100 000000 .WORD 0 ; (FILLED IN BY NTL)
121 000102 000000 .WORD 0 ; APR5 BIAS FOR LOADABLE DRIVERS
122
123 .IF DF M$$PRO

```

58  
59  
60  
61

:  
:  
:  
:

DECnet-Micro/R SX V1.0



|     |       |        |
|-----|-------|--------|
| 545 | .WORD | 27600  |
| 546 | .WORD | 167501 |
| 547 | .WORD | 26400  |
| 548 | .WORD | 166701 |
| 549 | .WORD | 166201 |
| 550 | .WORD | 26100  |
| 551 | .WORD | 162001 |
| 552 | .WORD | 22300  |
| 553 | .WORD | 22600  |
| 554 | .WORD | 162501 |
| 555 | .WORD | 23400  |
| 556 | .WORD | 163701 |
| 557 | .WORD | 163201 |
| 558 | .WORD | 23100  |
| 559 | .WORD | 21000  |
| 560 | .WORD | 161301 |
| 561 | .WORD | 161601 |
| 562 | .WORD | 21500  |
| 563 | .WORD | 160401 |
| 564 | .WORD | 20700  |
| 565 | .WORD | 20200  |
| 566 | .WORD | 160101 |
| 567 | .WORD | 120001 |
| 568 | .WORD | 60300  |
| 569 | .WORD | 60600  |
| 570 | .WORD | 120501 |
| 571 | .WORD | 61400  |
| 572 | .WORD | 121701 |
| 573 | .WORD | 121201 |
| 574 | .WORD | 61100  |
| 575 | .WORD | 63000  |
| 576 | .WORD | 123301 |
| 577 | .WORD | 123601 |
| 578 | .WORD | 63500  |
| 579 | .WORD | 122401 |
| 580 | .WORD | 62700  |
| 581 | .WORD | 62200  |
| 582 | .WORD | 122101 |
| 583 | .WORD | 66000  |
| 584 | .WORD | 126301 |
| 585 | .WORD | 126601 |
| 586 | .WORD | 66500  |
| 587 | .WORD | 127401 |
| 588 | .WORD | 67700  |
| 589 | .WORD | 67200  |
| 590 | .WORD | 127101 |
| 591 | .WORD | 125001 |
| 592 | .WORD | 65300  |
| 593 | .WORD | 65600  |
| 594 | .WORD | 125501 |
| 595 | .WORD | 64400  |
| 596 | .WORD | 124701 |
| 597 | .WORD | 124201 |
| 598 | .WORD | 64100  |
| 599 | .WORD | 74000  |
| 600 | .WORD | 134301 |
| 601 | .WORD | 134601 |

```
63 .SBTTL MACRO DEFINITIONS
64
65 .MCALL CCBDF$,NHWDF$,SAVRG,RESRG
66
67 CCBDF$; DEFINE CCB OFFSETS
68 000000 ; DEFINE HARDWARE REGISTERS
69 NHWDF$
70
71 ;
72 ; DEFINE LOCAL MACROS
73 ;
74 .IF DF P$$40!P$$34!P$$45!P$$70!L$$SI1!M$$MGE
75 R$$EIS=0 ; PROCESSOR HAS XOR INSTRUCTION
76
77 .ENDC
78
79 .IF NDF R$$EIS
80
81 .MACRO XOR,A,B
82 MOV A,R3 ; COPY A
83 BIC B,R3 ; A <- A AND NOT B
84 BIC A,B ; B <- B AND NOT A
85 BIS R3,B ; B <- A OR B
86 .ENDM XOR
87
88 .ENDC
```

|     |        |        |       |        |
|-----|--------|--------|-------|--------|
| 602 | 000742 | 074500 | .WORD | 74500  |
| 603 | 000744 | 135401 | .WORD | 135401 |
| 604 | 000746 | 075700 | .WORD | 75700  |
| 605 | 000750 | 075200 | .WORD | 75200  |
| 606 | 000752 | 135101 | .WORD | 135101 |
| 607 | 000754 | 137001 | .WORD | 137001 |
| 608 | 000756 | 077300 | .WORD | 77300  |
| 609 | 000760 | 077600 | .WORD | 77600  |
| 610 | 000762 | 137501 | .WORD | 137501 |
| 611 | 000764 | 076400 | .WORD | 76400  |
| 612 | 000766 | 136701 | .WORD | 136701 |
| 613 | 000770 | 136201 | .WORD | 136201 |
| 614 | 000772 | 076100 | .WORD | 76100  |
| 615 | 000774 | 132001 | .WORD | 132001 |
| 616 | 000776 | 072300 | .WORD | 72300  |
| 617 | 001000 | 072600 | .WORD | 72600  |
| 618 | 001002 | 132501 | .WORD | 132501 |
| 619 | 001004 | 073400 | .WORD | 73400  |
| 620 | 001006 | 133701 | .WORD | 133701 |
| 621 | 001010 | 133201 | .WORD | 133201 |
| 622 | 001012 | 073100 | .WORD | 73100  |
| 623 | 001014 | 071000 | .WORD | 71000  |
| 624 | 001016 | 131301 | .WORD | 131301 |
| 625 | 001020 | 131601 | .WORD | 131601 |
| 626 | 001022 | 071500 | .WORD | 71500  |
| 627 | 001024 | 130401 | .WORD | 130401 |
| 628 | 001026 | 070700 | .WORD | 70700  |
| 629 | 001030 | 070200 | .WORD | 70200  |
| 630 | 001032 | 130101 | .WORD | 130101 |
| 631 | 001034 | 050000 | .WORD | 50000  |
| 632 | 001036 | 110301 | .WORD | 110301 |
| 633 | 001040 | 110601 | .WORD | 110601 |
| 634 | 001042 | 050500 | .WORD | 50500  |
| 635 | 001044 | 111401 | .WORD | 111401 |
| 636 | 001046 | 051700 | .WORD | 51700  |
| 637 | 001050 | 051200 | .WORD | 51200  |
| 638 | 001052 | 111101 | .WORD | 111101 |
| 639 | 001054 | 113001 | .WORD | 113001 |
| 640 | 001056 | 053300 | .WORD | 53300  |
| 641 | 001060 | 053600 | .WORD | 53600  |
| 642 | 001062 | 113501 | .WORD | 113501 |
| 643 | 001064 | 052400 | .WORD | 52400  |
| 644 | 001066 | 112701 | .WORD | 112701 |
| 645 | 001070 | 112201 | .WORD | 112201 |
| 646 | 001072 | 052100 | .WORD | 52100  |
| 647 | 001074 | 116001 | .WORD | 116001 |
| 648 | 001076 | 056300 | .WORD | 56300  |
| 649 | 001100 | 056600 | .WORD | 56600  |
| 650 | 001102 | 116501 | .WORD | 116501 |
| 651 | 001104 | 057400 | .WORD | 57400  |
| 652 | 001106 | 117701 | .WORD | 117701 |
| 653 | 001110 | 117201 | .WORD | 117201 |
| 654 | 001112 | 057100 | .WORD | 57100  |
| 655 | 001114 | 055000 | .WORD | 55000  |
| 656 | 001116 | 115301 | .WORD | 115301 |
| 657 | 001120 | 115601 | .WORD | 115601 |
| 658 | 001122 | 055500 | .WORD | 55500  |

```
90 .SBTTL DEFINE KG-11 REGISTERS AND BITS
91
92 170700 KGCSR=170700 ; KG-11 CSR
93 000001 CRC16=1 ; USE CRC-16 POLYNOMIAL
94 000003 LRC16=3 ; USE LRC-16 POLYNOMIAL
95 000010 DDB=10 ; SET TO WORD MODE
96 000020 CLRKG=20 ; INITIALISE KG-11
97 000100 SEN=100 ; ENTER CYCLE MODE
98 000133 KGLDBC=DDB!LRC16!CLRKG!SFN ; LOAD NEW BCC ACCUMULATION
99 000111 KGINIT=DDB!CRC16!SEN ; INIT TO DO CRC-16 CALCULATION
```

|     |       |        |
|-----|-------|--------|
| 659 | .WORD | 114401 |
| 660 | .WORD | 54700  |
| 661 | .WORD | 54200  |
| 662 | .WORD | 114101 |
| 663 | .WORD | 104001 |
| 664 | .WORD | 44300  |
| 665 | .WORD | 44600  |
| 666 | .WORD | 104501 |
| 667 | .WORD | 45400  |
| 668 | .WORD | 105701 |
| 669 | .WORD | 105201 |
| 670 | .WORD | 45100  |
| 671 | .WORD | 47000  |
| 672 | .WORD | 107301 |
| 673 | .WORD | 107601 |
| 674 | .WORD | 47500  |
| 675 | .WORD | 106401 |
| 676 | .WORD | 46700  |
| 677 | .WORD | 46200  |
| 678 | .WORD | 106101 |
| 679 | .WORD | 42000  |
| 680 | .WORD | 102301 |
| 681 | .WORD | 102601 |
| 682 | .WORD | 42500  |
| 683 | .WORD | 103401 |
| 684 | .WORD | 43700  |
| 685 | .WORD | 43200  |
| 686 | .WORD | 103101 |
| 687 | .WORD | 101001 |
| 688 | .WORD | 41300  |
| 689 | .WORD | 41600  |
| 690 | .WORD | 101501 |
| 691 | .WORD | 40400  |
| 692 | .WORD | 100701 |
| 693 | .WORD | 100201 |
| 694 | .WORD | 40100  |
| 695 |       |        |
| 696 | .IFF  |        |
| 697 |       |        |
| 698 | .WORD | 0      |
| 699 | .WORD | 146001 |
| 700 | .WORD | 154001 |
| 701 | .WORD | 12000  |
| 702 | .WORD | 170001 |
| 703 | .WORD | 36000  |
| 704 | .WORD | 24000  |
| 705 | .WORD | 162001 |
| 706 | .WORD | 120001 |
| 707 | .WORD | 66000  |
| 708 | .WORD | 74000  |
| 709 | .WORD | 132001 |
| 710 | .WORD | 50000  |
| 711 | .WORD | 116001 |
| 712 | .WORD | 104001 |
| 713 | .WORD | 42000  |
| 714 |       |        |
| 715 | .ENDC |        |

;SECOND HALF OF TABLE

```

120
121
122
123 000076 006302
124 000100 067702 000000G
125 000104 011202
126
127
128
129 000106 032712 000007
130 000112 001746
131 000114 105312
132 000116 005377 000000G
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151 000122 010364 000006
152 000126 016205 000004
153 000132 116202 000002
154 000136 010403
155 000140 062703 000010
156 000144
157
158
159
160
161
162
163
164
165
166 000150
167
168
169
170 000156
171 000162 103413
172 000164 012764 001002 000010
173 000172 000716
174
175 000174 005704
176 000176 001402

.IFF
ASL R2 ; FORM WORD INDEX
ADD @SLTMA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE
MOV (R2),R2 ; GET START ADDRESS OF SLT ENTRY

.ENDC

BIT #LF.BWT,(R2) ; ANY WAIT REQUESTS HERE ?
BEQ 10$; IF EQ, NO - TRY NEXT LINE
DECB (R2) ; REDUCE NUMBER OF WAIT REQUESTS
DEC @RDQCT ; REDUCE NUMBER OF WAIT REQUESTS

.IF DF R$$MPL
.IF NDF R$$PRO
BIT #F2.MP,@FMSK2 ; IS THIS A MULTI PROCESSOR?
BEQ 22$; BR IF NO
MOV L.KRBA(R2),R5 ; GET KRB ADDRESS
MOV K.URM(R5),C.URM(R4) ; AND SET UP UNIBUS RUN MASK
CLRB C.MOD(R4) ; CHANGE SUB-FUNCTION CODE
MOV R3,C.NSP(R4) ; SAVE SYSTEM LINE NUMBER
MOV #100000,R3 ; SET UP DESTINATION PROCESS (AUX)
CALL @PDQUE ; AND RE-QUEUE ON CORRECT PROCESSOR
BR 25$

.ENDC

.IFTF
22$: MOV R3,C.LIN(R4) ; SET SYSTEM LINE NUMBER IN CCB
MOV L.DDS(R2),R5 ; SET PROCESS LINE TABLE ADDRESS
MOVB L.DDM(R2),R2 ; SET PROCESS PDV INDEX
MOV R4,R3 ; COMPUTE ADDRESS OF FUNCTION CODE
ADD #C.FNC,R3 ;
CALL @PDSPL ;: DISPATCH TO PROCESS

.ENDC

.IF DF R$$11D!1$:AS
MTPS #PR3 ; DROP BACK TO PRIORITY THREE

.IFF
25$: MTPS #PRO ; DROP BACK TO PRIORITY ZERO

.ENDC

CALL @RDBGT ; TRY FOR ANOTHER BUFFER
BCS 50$; IF CS, NO BUFFER - EXIT
MOV #FC.RCE+FS.RTN,C.FNC(R4)
BR 10$; TRY TO SATISFY ANY OTHER WAIT REQUESTS

30$: TST R4 ;: BUFFER TO RETURN ?
BEQ 40$;: IF EQ, NO - EXIT

```

|                 |                  |                 |                   |                    |
|-----------------|------------------|-----------------|-------------------|--------------------|
| ALOCB 000004RG  | G\$STTK= 000000  | PD\$PL 000072RG | T\$KMG= 000000    | \$PDDSP= ***** GX  |
| ASCPM 000032RG  | G\$SWRD= 000000  | PDVNM 000074RG  | T\$MIN= 000000    | \$PDQUE= ***** GX  |
| A\$CHK= 000000  | INTCT 000020RG   | PDVTA 000076RG  | TISCL 000146RG    | \$PDSPL= ***** GX  |
| A\$CPS= 000000  | I\$RAR= 000000   | PUMR 000100RG   | T100C 000150RG    | \$PDVNM= ***** GX  |
| A\$PRI= 000000  | I\$RDN= 000000   | PWRF1 000102RG  | T100Q 000152RG    | \$PDVTA= ***** GX  |
| A\$TRP= 000000  | KISAR6= ***** GX | P\$P45= 000000  | UMRPT 000030RG    | \$PUMR = ***** GX  |
| CCBGT 000034RG  | KSAR6 000002RG   | P\$WRD= 000000  | V\$CTR= 001000    | \$PWRF1= ***** GX  |
| CCBRT 000036RG  | K\$CNT= 177546   | Q\$TRT 000104RG | XAVL 000154RG     | \$Q\$TRT= ***** GX |
| CEACC 000040RG  | K\$CSR= 177546   | Q\$OPT= 000010  | X\$DBT= 000000    | \$RDBGT= ***** GX  |
| CEDIV 000042RG  | K\$SLD= 000000   | RDBGT 000106RG  | ZTIME 000156RG    | \$RDBNM= ***** GX  |
| CLINS 000006RG  | K\$TPS= 000074   | RDBNM 000110RG  | ZTIME2 000160RG   | \$RDBRT= ***** GX  |
| CMFRK 000044RG  | LD\$LP = 000000  | RDBRT 000112RG  | \$ALOCB= ***** GX | \$RDBSZ= ***** GX  |
| CXOPT 000046RG  | L\$ASG= 000000   | RDBSZ 000114RG  | \$ASCMP= ***** GX | \$RDBTH= ***** GX  |
| C\$CKP= 000000  | L\$DRV= 000000   | RDBTH 000116RG  | \$AUXVL= 000070 G | \$RDQCT= ***** GX  |
| C\$ORE= 000400  | L\$P11= 000001   | RDQCT 000120RG  | \$AUXVT 000000RG  | \$RDQSL= ***** GX  |
| C\$RSH= 177564  | L\$11R= 000000   | RDQSL 000122RG  | \$CCBGT= ***** GX | \$SLTMA= ***** GX  |
| DDAST 000050RG  | M\$CRB= 000124   | R\$DER= 000000  | \$CCBRT= ***** GX | \$SLTNM= ***** GX  |
| DDCCP 000052RG  | M\$CRX= 000000   | R\$K11= 000001  | \$CEACC= ***** GX | \$SORCM= ***** GX  |
| DDFNC 000054RG  | M\$FCS= 000000   | R\$SND= 000000  | \$CEDIV= ***** GX | \$SRSTD= ***** GX  |
| DDMSN 000056RG  | M\$MGE= 000000   | R\$11M= 000000  | \$CLINS= ***** GX | \$STDDT= ***** GX  |
| DEACB 000010RG  | M\$NET= 000000   | SLTMA 000124RG  | \$CMFRK= ***** GX | \$STDLC= ***** GX  |
| D\$PTM 000060RG | M\$OVR= 000000   | SLTNM 000126RG  | \$CXOPT= ***** GX | \$STDLL= ***** GX  |
| D\$BUG= 177514  | NMCLH 000062RG   | SQRCM 000130RG  | \$DDAST= ***** GX | \$STMFC= ***** GX  |
| D\$ISK= 000000  | NMCL2 000064RG   | SRSTD 000022RG  | \$DDCCP= ***** GX | \$TKPS = ***** GX  |
| D\$L11= 000001  | N\$AC= 000001    | STDD1 000132RG  | \$DDFNC= ***** GX | \$TK100= ***** GX  |
| D\$YNC= 000000  | N\$BUF= 000001   | STDLC 000134RG  | \$DDMSN= ***** GX | \$TSKRM= ***** GX  |
| D\$YNM= 000000  | N\$LDV= 000001   | STDL1 000136RG  | \$DEACB= ***** GX | \$TSTIM= ***** GX  |
| EXRQF 000012RG  | N\$MCP= 000001   | STMFC 000140RG  | \$DSPTM= ***** GX | \$TTISCL= ***** GX |
| EXRQN 000014RG  | N\$MML= 000001   | S\$WRG= 000000  | \$EXRQF= ***** GX | \$T100C= ***** GX  |
| E\$XPR= 000000  | N\$MOV= 000010   | S\$YSZ= 007600  | \$EXRQN= ***** GX | \$T100Q= ***** GX  |
| FMSK2 000016RG  | N\$NCT= 000001   | TKPS 000024RG   | \$FMASK= ***** GX | \$SUMRPT= ***** GX |
| F\$LVL= 000001  | N\$PEM= 000001   | TK100 000142RG  | \$INTCT= ***** GX | \$XAVL = ***** GX  |
| G\$TPP= 000000  | PDDSP 000066RG   | TSKRT 000026RG  | \$NMCLH= ***** GX | \$ZTIME= ***** GX  |
| G\$TSS= 000000  | PDOUE 000070RG   | T\$TIM 000144RG | \$NMCL2= ***** GX |                    |

. ABS. 000000 000 (RW,I,GBL,ABS,OVR)  
QC162 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 8914 Words ( 35 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:06.14  
SY:AXDAT.V2,[131,134]AXDAT/CR/-SP=SY:[1,1]RSXCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXCM/PA:1,[131,10]AXDAT

250  
251  
252

.ENDC

.ENDC



```

80 ;
81 ; AUXILARY PROCESS DISPATCH TABLE
82 ;
83 ;
84 000000 $AUXTB::
85 000000 .WORD +1 ; TRANSMIT ENABLE (NOP)
86 000002 .WORD $BFRN ; RECEIVE ENABLE
87 000004 .WORD +1 ; KILLIO (NOP)
88 .IF DF X$$MDC ;
89 000006 .WORD MDMCTL ; MODEM CONTROL CONTROL ROUTINE
90 .IF DF P$$RFL ;
91 000010 .WORD PWFAIL ; PERFORM POWERFAIL RECOVERY
92 .IFF
93 .WORD MDMSCN ; TIMEOUT ENTERS SCAN ROUTINE
94 .ENDC
95 .IFF
96 .WORD +1 ; CONTROL ENABLE (NOP)
97 .IF DF P$$RFL ;
98 .WORD PWFAIL ; PERFORM POWERFAIL RECOVERY
99 .IFF
100 .WORD DUMMY ; TIMEOUT (NOP)
101 .ENDC
102 .ENDC
103 000012 .WORD +1 ; TRANSMIT COMPLETE (NOP)
104 000014 .WORD +1 ; RECEIVE COMPLETE (NOP)
105 000016 .WORD +1 ; KILL COMPLETE (NOP)
106 000020 .WORD CCBRET ; CONTROL COMPLETE - RELEASE CCB
107 .IF DF P$$RFL ;
108 000022 .WORD PWRFL ; POWERFAIL DISPATCH TO DDM MODULES
109 .IFF
110 .WORD DUMMY ; POWERFAIL DISPATCH TO DDM MODULES (NOP)
111 .ENDC
112 000024 .WORD NMCMP ; NETWORK MANAGEMENT COUNTER COMPLETION
113
114 .IF NDF X$$MDC & R$$11D & I$$AS
115 $MDCIN::
116 .ENDC
117
118 DUMMY: RETURN
119 000026 CCBRET: CALLR @CCBRT
120 000030

```

AXDSPM - AUXILLIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 10:59<sup>L14</sup>  
Table of contents

10- 254 NETWORK MANAGEMENT COUNTER COMPLETION

AXDSPM CREATED BY MACRO ON 3-SEP-85 AT 10:59

PAGE 1 L 15

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL | VALUE      | REFERENCES    |
|--------|------------|---------------|
| CB.CCB | = 000002   | 10-306        |
| CCBRET | = 000030 R | 7-106 #7-120  |
| CCBRT  | = ***** GX | 7-120 10-308  |
| CS.LST | = 040000   | 10-303        |
| C.ADD  | 000034     | 10-270        |
| C.BID  | 000003     | 10-306        |
| C.CNT2 | 000030     | 10-274 10-286 |
| C.STS  | 000012     | *10-303       |
| DUMMY  | 000026 R   | 7-110 #7-119  |
| EXRON  | = ***** GX | 10-304        |
| ISSAS  | = *****    | 5-1 7-115     |
| LF.ACT | = 100000   | #6-77         |
| LF.BRO | = 000400   | #6-77 10-277  |
| LF.BWT | = 000007   | #6-77         |
| LF.ENA | = 002000   | #6-77         |
| LF.LPB | = 001000   | #6-77         |
| LF.MDC | = 000100   | #6-77         |
| LF.MFL | = 004000   | #6-77         |
| LF.MTP | = 000020   | #6-77 10-279  |
| LF.PAC | = 000200   | #6-77         |
| LF.RDY | = 040000   | #6-77         |
| LF.REA | = 010000   | #6-77         |
| LF.SER | = 000040   | #6-77 10-281  |
| LF.TIM | = 000010   | #6-77         |
| LF.UNL | = 020000   | #6-77         |
| LF.X2P | = 000000   | #6-77         |
| LN.CLO | = 000000   | #6-77         |
| LN.DUM | = 000005   | #6-77         |
| LN.LOA | = 000004   | #6-77         |
| LN.LOO | = 000003   | #6-77         |
| LN.OAU | = 000003   | #6-77         |
| LN.OFF | = 000001   | #6-77         |
| LN.ON  | = 000000   | #6-77         |
| LN.OOP | = 000004   | #6-77         |
| LN.OPE | = 000001   | #6-77         |
| LN.REF | = 000002   | #6-77         |
| LN.SER | = 000002   | #6-77         |
| LN.STA | = 000017   | #6-77         |
| LN.SUB | = 000360   | #6-77         |
| LN.TRI | = 000006   | #6-77         |
| L.COST | 000015     | #6-77         |
| L.CTL  | 000012     | #6-77         |
| L.CVA  | 177776     | #6-77         |
| L.DDM  | 000002     | #6-77         |
| L.DDS  | 000004     | #6-77         |
| L.DLC  | 000003     | #6-77         |
| L.DLM  | 000006     | #6-77         |
| L.DLS  | 000010     | #6-77         |
| L.FLG  | 000000     | #6-77         |
| L.KR54 | 000016     | #6-77         |
| L.LEN  | = 000022   | #6-77 10-285  |
| L.MPF  | 000022     | #6-77         |

```

254 .SBTTL NETWORK MANAGEMENT COUNTER COMPLETION
255
256 *
257 **--NMCMP-NETWORK MANAGEMENT COUNTER COMPLETION
258
259 THIS ROUTINE IS ENTERED WHEN A NETWORK MANAGEMENT REQUEST FOR COUNTERS
260 HAS BEEN COMPLETED BY A LOWER LEVEL PROCESS.
261 THE ROUTINE ALSO REQUESTS/UNSTOPS NETACP TO PROCESS
262 REQUESTS FOR LINE WATCHER
263
264 INPUTS:
265 R4 = ADDRESS OF COUNTER CCB
266
267 000304 054134 003310 NETACP: .RAD50 /NETACP/
268
269 000310 010403 000034 NMCMP: MOV R4,R3 ; Copy the ccb address
270 000312 062703 ; ADD #C.ADD,R3 ; Point to the task name
271 000316 005713 ; TST (R3) ; Is it for NETACP ?
272 000320 001041 ; BNE 40$; If NE, no
273
274 000322 116403 000030 MOVB C.CNT2(R4),R3 ; Copy SLN
275 000326 006303 ; ASL R3 ; Form word index
276 000330 067703 000000G ADD @SL7MA,R3 ; Point to line table address
277 000334 032773 000400 000000 BIT #LF.BRO,@(R3) ; Is it a broadcast channel?
278 000342 001026 ; BNE 30$; If NE, yes - EPM already filtered it.
279 000344 032773 000020 000000 BIT #LF.MTP,@(R3) ; Is it multipoint?
280 000352 001005 ; BNE 5$; If NE, yes - check station table
281 000354 032773 000040 000000 BIT #LF.SER,@(R3) ; Is service disabled ?
282 000362 001416 ; BEQ 30$; If EQ, no - allow request
283 000364 000437 ; BR 60$; Else, toss the request
284 000366 011303 5$: MOV (R3),R3 ; Get to the end of the line table
285 000370 062703 000022 ADD #L.LEN,R3 ; ...
286 000374 116401 000031 MOVB C.CNT2+1(R4),R1 ; Get the station number
287 000400 001404 ; BEQ 20$
288 000402 062703 000004 10$: ADD #S.LEN,R3 ; Move to the correct station table
289 000406 ; SOB R1,10$; ...
290 000412 032713 000001 20$: BIT #SF.SER,(R3) ; Is service disabled ?
291 000416 001022 ; BNE 60$; If NE, yes - toss the request
292
293 000420 012703 000304' 30$: MOV #NETACP,R3 ; Else start up NETACP
294 000424 40$: CALL @SRSTD ; Scan STD for task's TCB
295 000430 103415 ; BCS 60$; If CS, not there !!
296
297 000432 017703 000000G MOV @NMCL2,R3 ; Get address of net man listhead
298 000436 010413 ; MOV R4,(R3) ; Add CCB to end of head pointer
299 000440 010477 000000G 50$: MOV R4,@NMCL2 ; Update the tail pointer
300 000444 010403 ; MOV R4,R3 ; Copy possible 'last' buffer address
301 000446 011404 ; MOV (R4),R4 ; Else, get to the end of this chain
302 000450 001373 ; BNE 50$; If NE, get next buffer
303 000452 052763 040000 000012 BIS #CS.LST,C.STS(R3) ; Else, set end of chain flag
304 000460 ; CALLR @EXRON ; Request the task (unstop or request)
305
306 000464 122764 000002 000003 60$: CMPB #CB.CCB,C.BID(R4) ; Is it a CCB ?
307 000472 001002 ; BNE 70$; If NE, no
308 000474 ; CALLR @CCBRT ; Return the CCB
309 000500 70$: SAVRG <(R4)> ; Save the next in the chain
310 000502 ; CALL @RDBRT ; Return this one

```

```

365 .SBTTL $CEACC - ACCESS BLOCK IN EXTENDED POOL
366 .SBTTL $CECAC - ACCESS BLOCK IN ALTERNATE EXTENDED POOL
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390 000400 016767 000000G 000000G $CEACC::MOV $XBIAS,KISAR6 ; MAP TO BASE ADDRESS OF POOL
391 000406 032766 000001 000002 $CECAC::BIT #1,2(SP) ; IS THE BLOCK IN EXTENDED POOL ?
392 000414 001420 BEQ 100$; IF EQ, NO - ALREADY MAPPED TO IT
393 000416 SAVRG <R0> ; ELSE, SAVE R0
394 000420 016600 000004 MOV 4(SP),R0 ; COPY BLOCK ADDRESS
395 000424 042766 177701 000004 BIC #177701,4(SP) ; RETURN VIRTUAL ADDRESS
396 000432 052766 140000 000004 BIS #140000,4(SP) ; ... MAPPED THROUGH KISAR6
397
398
399
400 000440 072027 177772
401
402
403
404
405
406
407
408
409
410 000444 042700 176000 BIC #176000,R0 ; CLEAR EXTRANEIOUS BITS
411 000450 060067 000000G ADD R0,KISAR6 ; ... UPDATE TO PROPER BLOCK
412 000454 RESRG <R0> ; RESTORE R0
413 000456 100$: RETURN ; RETURN TO THE CALLER

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

.TITLE CETIM  
.IDENT /V05.00/

COPYRIGHT (C) 1978,1979,1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A  
SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR  
ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE  
MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH  
SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE  
TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN  
IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT  
NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL  
EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF  
ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

#### MODULE DESCRIPTION

CEX TIMER SERVICE ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

#### IDENT HISTORY:

2.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0  
  
3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1  
  
4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0  
  
5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

.MCALL INHIB\$,ENABL\$,SAVRG,RESRG  
.MCALL CCBDF\$,SLTDF\$,PDVDF\$  
CCBDF\$ ; DEFINE THE CCB OFFSETS  
SLTDF\$ ; DEFINE THE SLT OFFSETS

00000L  
00000L

```

124 .WORD 0 ; Unibus run mask
125 .ENDC
126
127 $LTMFC::
128 .IF NDF R$$MPL
129 .WORD FC.TIM+FS.LTM ; FUNCTION CODE TO DISPATCH LONG TIMEOUTS
130 .IFF ; NDF R$$MPL
131 .WORD 0
132 .ENDC ; NDF R$$MPL
133
134 000106 000000 000000 000000 $T1SCL:: .WORD 0,0,0,0,0 ; 1 SECOND CLOCK QUEUE ENTRY
135 000114 000000 000000
136 000120 000000
137 .WORD 0 ; ADDRESS OF PROCESSING ROUTINE
138 .WORD 0 ; (FILLED IN BY NTL)
139 .WORD 0 ; APR5 BIAS FOR LOADABLE DRIVERS
140 .IF DF M$$PRO
141 .WORD 0 ; Unibus run mask
142 .ENDC
143 000124 000000 $DECP:: .WORD 0 ; Pointer to DECnet Home Block
144 000126 000000 $PSIPT:: .WORD 0 ; POINTER TO PSI HOME BLOCK
145 000130 000000 $SNAPT:: .WORD 0 ; POINTER FOR SNA
146
147 000132 $DDFNC::
148 .IF NDF R$$MPL
149 .WORD FC.PWR ; POWERFAIL RECOVERY FUNCTION CODE
150 .IFF ; NDF R$$MPL
151 .WORD 0
152 .ENDC ; NDF R$$MPL
153
154 000134 $CCBLH:: .BLKW 1 ; POINTER TO FIRST FREE CCB
155 000136 $SDBCT:: .BLKW 1 ; COUNTER OF CURRENT SDB'S IN POOL
156 000140 $SDBLH:: .BLKW 1 ; POINTER TO FIRST FREE SMALL DATA BUFFER
157 000142 .BLKW 1 ; SECOND HALF OF ADDRESS DOUBLEWORD
158 000144 $SDBAF:: .BLKW 1 ; COUNTER OF # OF SDB ALLOCATION FAILURES
159 000146 $RDBCT:: .BLKW 1 ; COUNTER OF CURRENT RDB'S IN POOL
160 000150 $RDBLH:: .BLKW 1 ; POINTER TO FIRST FREE RECEIVE BUFFER
161 000152 .BLKW 1 ; SECOND HALF OF ADDRESS DOUBLEWORD
162 000154 $RDBAF:: .BLKW 1 ; COUNTER OF # OF RDB ALLOCATION FAILURES
163 000156 $RDQSL:: .BLKW 1 ; SYSTEM LINE # TO CHECK ON NEXT BUFFER RETURN
164 000160 $RDQCT:: .BLKW 1 ; NUMBER OF BUFFER WAIT REQUESTS AND FLAG
165 000162 $NTLPT:: .BLKW 1 ; NTL POINTER TO IMPURE AREA IN NTPool
166 000164 $CXOPT:: .BLKW 1 ; COMM/EXEC OPTIONS INCLUDED
167
168 000166 $NBIAS:: .BLKW 1 ; POINTER TO NON-UMR MAPPED POOL
169 000170 $QBIAS:: .BLKW 1 ; POINTER TO UMR-MAPPED PORTION OF POOL
170 000172 $QSTRT:: .BLKW 1 ; ORIGINAL BIAS OF UMR-MAPPED PORTION OF POOL
171 .IF NDF R$$MPL
172 000174 $PUMR:: .BLKW 1 ; STARTING UMR BIAS
173 .ENDC
174 000176 $PWRF1:: .BLKW 1 ; POWER FAIL FLAG
175 000200 $XBIAS:: .BLKW 1 ; BIAS OF EXTENDED SINGLE WORD POOL
176 000202 $PAVL:: ; PHASE III COMPATIBILITY
177 000202 000000 $XAVL:: .WORD 0 ; POINTER TO FIRST FREE BLOCK IN EXTENDED POOL
178 000204 000000 $NMLST:: .WORD 0 ; Logical Name Listhead
179 000206 $OBJHD:: .BLKW 1 ; OBJECT TABLE LISTHEAD

```

```

63 .SBTTL MACRO DEFINITIONS
64
65 .MCALL CCBDF$,NHWDF$,SAVRG,RESRG
66
67 CCBDF$; DEFINE CCB OFFSETS
68 NHWDF$; DEFINE HARDWARE REGISTERS
69
70 ;
71 ; DEFINE LOCAL MACROS
72 ;
73 .IF DF P$$40!P$$34!P$$45!P$$70!L$$S11!M$$MGE
74
75 R$$EIS=0 ; PROCESSOR HAS XOR INSTRUCTION
76
77 .ENDC
78
79 .IF NDF R$$EIS
80
81 .MACRO XOR,A,B
82 MOV A,R3 ; COPY A
83 BIC B,R3 ; A <- A AND NOT B
84 BIC A,B ; B <- B AND NOT A
85 BIS R3,B ; B <- A OR B
86 .ENDM XOR
87
88 .ENDC

```



|     |       |        |
|-----|-------|--------|
| 602 | .WORD | 74500  |
| 603 | .WORD | 135401 |
| 604 | .WORD | 75700  |
| 605 | .WORD | 75200  |
| 606 | .WORD | 135101 |
| 607 | .WORD | 137001 |
| 608 | .WORD | 77300  |
| 609 | .WORD | 77600  |
| 610 | .WORD | 137501 |
| 611 | .WORD | 76400  |
| 612 | .WORD | 136701 |
| 613 | .WORD | 136201 |
| 614 | .WORD | 76100  |
| 615 | .WORD | 132001 |
| 616 | .WORD | 72300  |
| 617 | .WORD | 72600  |
| 618 | .WORD | 132501 |
| 619 | .WORD | 73400  |
| 620 | .WORD | 133701 |
| 621 | .WORD | 133201 |
| 622 | .WORD | 73100  |
| 623 | .WORD | 71000  |
| 624 | .WORD | 131301 |
| 625 | .WORD | 131601 |
| 626 | .WORD | 71500  |
| 627 | .WORD | 130401 |
| 628 | .WORD | 70700  |
| 629 | .WORD | 70200  |
| 630 | .WORD | 130101 |
| 631 | .WORD | 50000  |
| 632 | .WORD | 110301 |
| 633 | .WORD | 110601 |
| 634 | .WORD | 50500  |
| 635 | .WORD | 111401 |
| 636 | .WORD | 51700  |
| 637 | .WORD | 51200  |
| 638 | .WORD | 111101 |
| 639 | .WORD | 113001 |
| 640 | .WORD | 53300  |
| 641 | .WORD | 53600  |
| 642 | .WORD | 113501 |
| 643 | .WORD | 52400  |
| 644 | .WORD | 112701 |
| 645 | .WORD | 112201 |
| 646 | .WORD | 52100  |
| 647 | .WORD | 116001 |
| 648 | .WORD | 56300  |
| 649 | .WORD | 56600  |
| 650 | .WORD | 116501 |
| 651 | .WORD | 57400  |
| 652 | .WORD | 117701 |
| 653 | .WORD | 117201 |
| 654 | .WORD | 57100  |
| 655 | .WORD | 55000  |
| 656 | .WORD | 115301 |
| 657 | .WORD | 115601 |
| 658 | .WORD | 55500  |

STCRCE MACRO V05.03b Friday 28-Jun-85 18:22 Page 7  
 DEFINE KG-11 REGISTERS AND BITS

```

90
91
92 170700 KGCSR=170700 ; KG-11 CSR
93 000001 CRC16=1 ; USE CRC-16 POLYNOMIAL
94 000003 LRC16=3 ; USE LRC-16 POLYNOMIAL
95 000010 DDB=10 ; SET TO WORD MODE
96 000020 CLRKG=20 ; INITIALISE KG-11
97 000100 SEN=100 ; ENTER CYCLE MODE
98 000133 KGLDBC=DDB!LRC16!CLRKG!SFN ; LOAD NEW BCC ACCUMULATION
99 000111 KGINIT=DDB!CRC16!SEN ; INIT TO DO CRC-16 CALCULATION

```

|     |        |        |       |        |
|-----|--------|--------|-------|--------|
| 659 | 001124 | 114401 | .WORD | 114401 |
| 660 | 001126 | 054700 | .WORD | 54700  |
| 661 | 001130 | 054200 | .WORD | 54200  |
| 662 | 001132 | 114101 | .WORD | 114101 |
| 663 | 001134 | 104001 | .WORD | 104001 |
| 664 | 001136 | 044300 | .WORD | 44300  |
| 665 | 001140 | 044600 | .WORD | 44600  |
| 666 | 001142 | 104501 | .WORD | 104501 |
| 667 | 001144 | 045400 | .WORD | 45400  |
| 668 | 001146 | 105701 | .WORD | 105701 |
| 669 | 001150 | 105201 | .WORD | 105201 |
| 670 | 001152 | 045100 | .WORD | 45100  |
| 671 | 001154 | 047000 | .WORD | 47000  |
| 672 | 001156 | 107301 | .WORD | 107301 |
| 673 | 001160 | 107601 | .WORD | 107601 |
| 674 | 001162 | 047500 | .WORD | 47500  |
| 675 | 001164 | 106401 | .WORD | 106401 |
| 676 | 001166 | 046700 | .WORD | 46700  |
| 677 | 001170 | 046200 | .WORD | 46200  |
| 678 | 001172 | 106101 | .WORD | 106101 |
| 679 | 001174 | 042000 | .WORD | 42000  |
| 680 | 001176 | 102301 | .WORD | 102301 |
| 681 | 001200 | 102601 | .WORD | 102601 |
| 682 | 001202 | 042500 | .WORD | 42500  |
| 683 | 001204 | 103401 | .WORD | 103401 |
| 684 | 001206 | 043700 | .WORD | 43700  |
| 685 | 001210 | 043200 | .WORD | 43200  |
| 686 | 001212 | 103101 | .WORD | 103101 |
| 687 | 001214 | 101001 | .WORD | 101001 |
| 688 | 001216 | 041300 | .WORD | 41300  |
| 689 | 001220 | 041600 | .WORD | 41600  |
| 690 | 001222 | 101501 | .WORD | 101501 |
| 691 | 001224 | 040400 | .WORD | 40400  |
| 692 | 001226 | 100701 | .WORD | 100701 |
| 693 | 001230 | 100201 | .WORD | 100201 |
| 694 | 001232 | 040100 | .WORD | 40100  |
| 695 |        |        |       |        |
| 696 |        |        | .IFF  |        |
| 697 |        |        |       |        |
| 698 |        |        | .WORD | 0      |
| 699 |        |        | .WORD | 146001 |
| 700 |        |        | .WORD | 154001 |
| 701 |        |        | .WORD | 12000  |
| 702 |        |        | .WORD | 170001 |
| 703 |        |        | .WORD | 36000  |
| 704 |        |        | .WORD | 24000  |
| 705 |        |        | .WORD | 162001 |
| 706 |        |        | .WORD | 120001 |
| 707 |        |        | .WORD | 66000  |
| 708 |        |        | .WORD | 74000  |
| 709 |        |        | .WORD | 132001 |
| 710 |        |        | .WORD | 50000  |
| 711 |        |        | .WORD | 116001 |
| 712 |        |        | .WORD | 104001 |
| 713 |        |        | .WORD | 42000  |
| 714 |        |        |       |        |
| 715 |        |        | .ENDC |        |

;SECOND HALF OF TABLE

STCRCK MACRO V05.03b Friday 28-Jun-85 18:22 Page 8  
EXECUTIVE VECTOR TABLE

101

102

103 000000 000000

104 000002 000000G

105 000004 172354

106

107 000002

.SBTTL EXECUTIVE VECTOR TABLE

\$STCVT::WORD 0 ; FLAGS WORD

KSAR5: .WORD KISAR5 ;

KSAR6: .WORD KISAR6 ;

\$STCVL == &lt;&lt;.-\$STCVT&gt;/2&gt;-1 ; LENGTH OF VECTOR TABLE

STCRCK - MACRO V05.03b Friday 28-Jun-85 18:22 Page 12-5  
STCR2 - CALCULATE CRC

M 9

716  
717  
718  
719

000001

.ENDC

.END

```

177
178 000200 CALL @RDBRT ;;; RETURN BUFFER TO POOL
179
180 000204 40$: .IF DF R$$11D!I$$AS
181
182 MTPS #PR3 ; ENABLE INTERRUPTS
183
184 .IFF
185
186 000204 MTPS #PRO ; ENABLE INTERRUPTS
187
188 .ENDC
189
190 000212 50$: .IF DF R$$MPL
191 .IF NDF R$$PRO
192
193 BIT #F2.MP,@FMSK2 ; IS THIS A MULTIPROCESSOR?
194 BEQ 60$
195 CALL @($P)+ ; RESTORE STATE OF THE CACHE
196
197 60$:
198 .ENDC
199 .ENDC
200 000212 RETURN ; RETURN
201
202 .IF DF R$$MPL
203 .IF NDF R$$PRO
204
205 ;+
206 ;**--MPBRTN-BUFFER RETURN ROUTINE FOR MULTI-PROCESSORS
207 ;
208 ; THIS ROUTINE WILL BE EXECUTED ON THE PROCESSOR ON WHICH THE DEVICE
209 ; BEING GIVEN THE BUFFER IS CONNECTED.
210 ; -
211 ; INPUTS:
212 ; R4 - ADDRESS OF CCB/RDB TO SATISFY THE REQUEST
213 ;
214 MPBRTN: MOV (.NSP(R4),R5 ; GET SYSTEM LINE #
215 MOV R5,C.LIN(R4) ; SET LINE # IN CCB
216 ASL R5 ; COMPUTE ADDRESS OF
217 ADD @SLTMA,R5 ; SYSTEM LINE TABLE
218 MOV (R5),R5
219 MOV L.DDM(R5),R2 ; GET DDM PROCESS INDEX
220 MOV L.DDS(R5),R5 ; GET PROCESS LINE TABLE ADDRESS
221 MOV R4,R3 ; COPY CCB ADDRESS
222 ADD #C.FNC,R3 ; POINT TO FUNCTION CODE
223 MOV #FC.RCE+FS.RTN,(R3) ; RESET FUNCTION CODE
224 CALLR @PDSPL ; DISPATCH TO DDM PROCESS
225
226 .ENDC
227 .ENDC
228
229 000001 .END

```

AXDAT CREATED BY MACRO ON 28-JUN-85 AT 18:28

PAGE 1 M 11

SYMBOL CROSS REFERENCE

CREF 04.00

SYMBOL VALUE REFERENCES

|         |         |    |        |       |      |      |       |
|---------|---------|----|--------|-------|------|------|-------|
| ALOCB   | 000004  | RG | #5-50  |       |      |      |       |
| ASCOMP  | 000032  | RG | #5-86  |       |      |      |       |
| CCBGT   | 000034  | RG | #5-87  |       |      |      |       |
| CCBRT   | 000036  | RG | #5-88  |       |      |      |       |
| CEACC   | 000040  | RG | #5-89  |       |      |      |       |
| CEDIV   | 000042  | RG | #5-90  |       |      |      |       |
| CLINS   | 000006  | RG | #5-52  |       |      |      |       |
| CMFRK   | 000044  | RG | #5-91  |       |      |      |       |
| CXOPT   | 000046  | RG | #5-92  |       |      |      |       |
| DDAST   | 000050  | RG | #5-93  |       |      |      |       |
| DDCCP   | 000052  | RG | #5-94  |       |      |      |       |
| DDFNC   | 000054  | RG | #5-95  |       |      |      |       |
| DDMSN   | 000056  | RG | #5-96  |       |      |      |       |
| DEACB   | 000010  | RG | #5-60  |       |      |      |       |
| DSPTM   | 000060  | RG | #5-97  |       |      |      |       |
| EXRQF   | 000012  | RG | #5-61  |       |      |      |       |
| EXRON   | 000014  | RG | #5-62  |       |      |      |       |
| FMSK2   | 000016  | RG | #5-63  |       |      |      |       |
| INTCT   | 000020  | RG | #5-72  |       |      |      |       |
| KISAR6  | = ***** | GX | 5-49   |       |      |      |       |
| KSAR6   | 000002  | RG | #5-49  |       |      |      |       |
| NMCLH   | 000062  | RG | #5-105 |       |      |      |       |
| NMCL2   | 000064  | RG | #5-106 |       |      |      |       |
| PDDSP   | 000066  | RG | #5-111 |       |      |      |       |
| PDQUE   | 000070  | RG | #5-112 |       |      |      |       |
| PDSPL   | 000072  | RG | #5-113 |       |      |      |       |
| PDVNM   | 000074  | RG | #5-114 |       |      |      |       |
| PDVTA   | 000076  | RG | #5-115 |       |      |      |       |
| PUMR    | 000100  | RG | #5-116 |       |      |      |       |
| PWRF1   | 000102  | RG | #5-117 |       |      |      |       |
| QSTRT   | 000104  | RG | #5-118 |       |      |      |       |
| RDBG1   | 000106  | RG | #5-119 |       |      |      |       |
| RDBNM   | 000110  | RG | #5-120 |       |      |      |       |
| RDBRT   | 000112  | RG | #5-121 |       |      |      |       |
| RDBSZ   | 000114  | RG | #5-122 |       |      |      |       |
| RDBTH   | 000116  | RG | #5-123 |       |      |      |       |
| RDGCT   | 000120  | RG | #5-124 |       |      |      |       |
| RDQSL   | 000122  | RG | #5-125 |       |      |      |       |
| R\$EMPL | = ***** |    | 5-54   | 5-63  | 5-78 | 5-99 | 5-133 |
| R\$PRO  | = ***** |    | 5-106  | 5-145 |      |      | 5-151 |
| SLTMA   | 000124  | RG | #5-126 |       |      |      |       |
| SLTNM   | 000126  | RG | #5-127 |       |      |      |       |
| SQRCM   | 000130  | RG | #5-128 |       |      |      |       |
| SRSTD   | 000022  | RG | #5-73  |       |      |      |       |
| STDD1   | 000132  | RG | #5-129 |       |      |      |       |
| STDLC   | 000134  | RG | #5-130 |       |      |      |       |
| STDL1   | 000136  | RG | #5-131 |       |      |      |       |
| STMFC   | 000140  | RG | #5-132 |       |      |      |       |
| TKPS    | 000024  | RG | #5-74  |       |      |      |       |
| TK100   | 000142  | RG | #5-138 |       |      |      |       |
| TSKRT   | 000026  | RG | #5-75  |       |      |      |       |
| TSTIM   | 000144  | RG | #5-139 |       |      |      |       |

```

254 .SBTTL NETWORK MANAGEMENT COUNTER COMPLETION
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310

```

\*\*\*-NMCMP-NETWORK MANAGEMENT COUNTER COMPLETION  
 THIS ROUTINE IS ENTERED WHEN A NETWORK MANAGEMENT REQUEST FOR COUNTERS  
 HAS BEEN COMPLETED BY A LOWER LEVEL PROCESS.  
 THE ROUTINE ALSO REQUESTS/UNSTOPS NETACP TO PROCESS  
 REQUESTS FOR LINE WATCHER  
 -  
 INPUTS:  
 R4 = ADDRESS OF COUNTER CCB

```

267 000034 054134 003310 NETACP: .RAD50 /NETACP/
268
269 000040 010403 NMCMP: MOV R4,R3 ; Copy the ccb address
270 000042 062703 ADD #C.ADD,R3 ; Point to the task name
271 000046 005713 TST (R3) ; Is it for NETACP ?
272 000050 001041 BNE 40$; If NE, no
273
274 000052 116403 MOV B C.CNT2(R4),R3 ; Copy SLN
275 000056 006303 ASL R3 ; Form word index
276 000060 067703 ADD @SLTMA,R3 ; Point to line table address
277 000064 032773 BIT #LF.BRO,@(R3) ; Is it a broadcast channel?
278 000072 001026 BNE 30$; If NE, yes - EPM already filtered it.
279 000074 032773 BIT #LF.MTP,@(R3) ; Is it multipoint?
280 000102 001005 BNE 5$; If NE, yes - check station table
281 000104 032773 BIT #LF.SER,@(R3) ; Is service disabled ?
282 000112 001416 BEQ 30$; If EQ, no - allow request
283 000114 000437 BR 60$; Else, toss the request
284 000116 011303 5$: MOV (R3),R3 ; Get to the end of the line table
285 000120 062703 ADD #L.LEN,R3 ; ...
286 000124 116401 MOV B C.CNT2+1(R4),R1 ; Get the station number
287 000130 001404 BEQ 20$
288 000132 062703 10$: ADD #S.LEN,R3 ; Move to the correct station table
289 000136 001404 SOB R1,10$; ...
290 000142 032713 20$: BIT #SF.SER,(R3) ; Is service disabled ?
291 000146 001022 BNE 60$; If NE, yes - toss the request
292
293 000150 012703 30$: MOV #NETACP,R3 ; Else start up NETACP
294 000154 103415 40$: CALL @SRSTD ; Scan STD for task's TCB
295 000160 103415 BCS 60$; If CS, not there !!
296
297 000162 017703 MOV @NMCL2,R3 ; Get address of net man listhead
298 000166 010413 MOV R4,(R3) ; Add CCB to end of head pointer
299 000170 010477 50$: MOV R4,@NMCL2 ; Update the tail pointer
300 000174 010403 MOV R4,R3 ; Copy possible 'last' buffer address
301 000176 011404 MOV (R4),R4 ; Else, get to the end of this chain
302 000200 001373 BNE 50$; If NE, get next buffer
303 000202 052763 BIS #CS.LST,C.STS(R3) ; Else, set end of chain flag
304 000210 000012 CALLR @EXRON ; Request the task (unstop or request)
305
306 000214 122764 60$: CMPB #CB.CCB,C.BID(R4) ; Is it a CCB ?
307 000222 001002 BNE 70$; If NE, no
308 000224 000003 CALLR @CCBRT ; Return the CCB
309 000230 SAVRG <(R4)> ; Save the next in the chain
310 000232 CALL @RDBRT ; Return this one

```



122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178

.IF DF P\$\$\$RFL  
.SBTTL POWERFAIL RECOVERY ROUTINE

\*\*\*PWFAIL-POWERFAIL RECOVERY ROUTINE

THIS ROUTINE IS INVOKED ONCE PER SECOND BY THE TIMER SERVICE CODE. IF THE POWERFAIL RECOVERY FLAG IS SET, WE WILL SCAN THE SYSTEM LINE TABLE FOR 'ACTIVE' LINES (I.E. LINES WHICH HAVE BOTH DLC AND DDM PROCESSES LOADED) AND ASYNCHRONOUSLY QUEUE A CONTROL COMPLETION TO THE LLC LEVEL INDICATING THAT THE LINK HAS BEEN DISCONNECTED.

```

PWFAIL: MOV @PWRF1,R1 ; GET # OF LINES REMAINING TO BE POWERFAILED
 BLE 100$; IF NONE ... NO RECOVERY UNDERWAY
 DEC R1 ; CONVERT TO SYSTEM LINE #
 MOV R1,R3 ; SAVE FOR LATER CALL TO $ASCMP

 .IF DF N$$$1LN
 MOV @SLTMA,R1 ; GET ADDRESS OF SYSTEM LINE TABLE
 MOV (R1),R1 ; ...

 .IFF
 ASL R1 ; FORM WORD INDEX
 ADD @SLTMA,R1 ; POINT INTO SYSTEM LINE MAPPING TABLE
 MOV (R1),R1 ; GET ADDRESS OF SYSTEM LINE TABLE

 .ENDC

 BIT #LF.ACT,(R1) ; IS THIS LINE 'ACTIVE'?
 BEQ 20$; NO ... NO RECOVERY REQUIRED
 CLR R2 ; ASSUME LINE IS NOT MULTIPOINT
 TSTB L.NSTA(R1) ; IS THIS LINE MULTIPOINT?
 BEQ 10$; IF EQ, NO
 MOVB TRIB+1,R2 ; GET TRIBUTARY NUMBER TO CHECK
 ASL R2 ; FORM DOUBLE WORD OFFSET
 ASL R2
 ADD R1,R2 ; POINT INTO SYSTEM LINE TABLE
 BITB #SF.ACT,L.MPF(R2) ; IS THE TRIBUTARY ACTIVE?
 BEQ 15$; NO ... NO RECOVERY REQUIRED
 MOV TRIB,R2 ; YES ... GET TRIBUTARY ADDRESS
 BISB R3,R2 ; SET SYSTEM LINE NUMBER
 MOV #CE.DIS,R3 ; YES ... SET UP ERROR CODE
 CALL @ASCMP ; PERFORM ASYNCHRONOUS COMPLETION
 BCS 100$; TRY LATER ON RESOURCE ALLOCATION FAILURE
 INCB TRIB+1 ; UPDATE TRIBUTARY ADDRESS
 CMPB TRIB+1,L.NSTA(R1) ; HAVE WE CHECKED ALL TRIBUTARIES ON THIS LINE?
 BLO PWFAIL ; IF LO, NO
 CLR TRIB ; RESET TRIBUTARY ADDRESS
 DEC @PWRF1 ; ONE LESS LINE TO RECOVER
 BNE PWFAIL ; LOOP TILL ALL DONE

10$: MOV #CE.DIS,R3 ; YES ... SET UP ERROR CODE
 CALL @ASCMP ; PERFORM ASYNCHRONOUS COMPLETION
 BCS 100$; TRY LATER ON RESOURCE ALLOCATION FAILURE
 INCB TRIB+1 ; UPDATE TRIBUTARY ADDRESS
 CMPB TRIB+1,L.NSTA(R1) ; HAVE WE CHECKED ALL TRIBUTARIES ON THIS LINE?
 BLO PWFAIL ; IF LO, NO
 CLR TRIB ; RESET TRIBUTARY ADDRESS
 DEC @PWRF1 ; ONE LESS LINE TO RECOVER
 BNE PWFAIL ; LOOP TILL ALL DONE

15$: MOV #CE.DIS,R3 ; YES ... SET UP ERROR CODE
 CALL @ASCMP ; PERFORM ASYNCHRONOUS COMPLETION
 BCS 100$; TRY LATER ON RESOURCE ALLOCATION FAILURE
 INCB TRIB+1 ; UPDATE TRIBUTARY ADDRESS
 CMPB TRIB+1,L.NSTA(R1) ; HAVE WE CHECKED ALL TRIBUTARIES ON THIS LINE?
 BLO PWFAIL ; IF LO, NO
 CLR TRIB ; RESET TRIBUTARY ADDRESS
 DEC @PWRF1 ; ONE LESS LINE TO RECOVER
 BNE PWFAIL ; LOOP TILL ALL DONE

20$: DEC @PWRF1 ; ONE LESS LINE TO RECOVER
 BNE PWFAIL ; LOOP TILL ALL DONE

100$: .IF DF X$$$MDC

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

```
.IF DF $$$1D!$$$AS
.IF DF $$$MDC
.TITLE AXDSPM - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSP - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.ENDC
.IFF
.IF DF $$$MDC
.IF DF $$$RFL
.TITLE AXDSPB - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSPM - AUXILLIARY PROCESS DISPATCH
.ENDC
.IFF
.IF DF $$$RFL
.TITLE AXDSPP - AUXILLIARY PROCESS DISPATCH AND POWER FAIL
.IFF
.TITLE AXDSP - AUXILLIARY PROCESS DISPATCH
.ENDC
.ENDC
.ENDC
.IDENT /V05.01/
```

COPYRIGHT (C) 1978,1979,1980, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

#### MODULE DESCRIPTION

AUXILARY PROCESS DISPATCH TABLE

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

#### IDENT HISTORY:

1.00 10-FEB-78  
VERSION 2.0 RELEASE  
2.00 14-DEC-79

AXDSPM CREATED BY MACRO ON 3-SEP-85 AT 10:59

PAGE 2 M 15

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE       | REFERENCES           |
|---------|-------------|----------------------|
| L.NMST  | 000020      | #6-77                |
| L.NSTA  | 000014      | #6-77                |
| L.OWNR  | 000021      | #6-77                |
| L.UNT   | 000013      | #6-77                |
| MDMCTL  | = ***** GX  | 7-89                 |
| MDMSCN  | = ***** GX  | 7-93                 |
| NETACP  | = 000034 R  | #10-267 10-293       |
| NMCL2   | = ***** GX  | 10-297 10-299        |
| NMCMP   | = 000040 R  | 7-112 #10-269        |
| PSSRFL  | = *****     | 5-9 7-90 7-107 8-122 |
| RDBRT   | = ***** GX  | 10-310               |
| RSS11D  | = *****     | 5-1 7-115            |
| SF.ACT  | = 000200    | #6-77                |
| SF.ENA  | = 000100    | #6-77                |
| SF.LPB  | = 000034    | #6-77                |
| SF.MFL  | = 000040    | #6-77                |
| SF.PAC  | = 000020    | #6-77                |
| SF.REA  | = 000010    | #6-77                |
| SF.SER  | = 000001    | #6-77 10-290         |
| SF.SVC  | = 000002    | #6-77                |
| SF.UNL  | = 000040    | #6-77                |
| SLTMA   | = ***** GX  | 10-276               |
| SRSTD   | = ***** GX  | 10-294               |
| S.COST  | = 000001    | #6-77                |
| S.FLG   | = 000000    | #6-77                |
| S.LEN   | = 000004    | #6-77 10-288         |
| S.NMST  | = 000002    | #6-77                |
| S.OWNR  | = 000003    | #6-77                |
| X\$MDC  | = 000001    | #4-2 5-8 7-88 7-115  |
| \$AUXTB | = 000000 RG | #7-84                |
| \$BFRTN | = ***** GX  | 7-86                 |

AXDSPP - AUXILLIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 11:00 <sup>M 16</sup> Page 10-1  
NETWORK MANAGEMENT COUNTER COMPLETION

311 000506  
312 000510 001373  
313 000512  
314  
315 000001

RESRG <R4> ; Recover the next buffer  
BNE 70\$ ; If NE, get next buffer  
RETURN ; Else, return  
END

```

415 .SBTTL $PDVID - PROCESS NAME TO PDV INDEX
416
417 ;+
418 **-- $PDVID - PROCESS NAME TO PDV INDEX
419 ;
420 THIS SUBROUTINE MAPS A PROCESS NAME (UP TO THREE CHARACTERS IN RAD50)
421 INTO A PDV INDEX.
422 ;
423 INPUTS:
424 R2 = PROCESS NAME (RAD50)
425 ;
426 OUTPUTS:
427 R2 = PDV INDEX
428 ;
429 -
430 000460 $PDVID::SAVRG <R0,R1> ; SAVE REGISTERS
431 000464 016700 000000G MOV $PDVTA,R0 ; GET ADDRESS OF PDV ADDRESS TABLE
432 000470 016746 000000G MOV $PDVNM, (SP) ; GET TOTAL NUMBER OF PDVS IN SYSTEM
433 000474 012001 10$: MOV (R0)+, ; GET ADDRESS OF PDV
434 000476 001403 BEQ 15$; PROCESS NOT LOADED - GO TO NEXT ONE
435 000500 026102 000004 CMP Z,NAM(R1),R2 ; DOES PRUCESS NAME MATCH?
436 000504 001404 BEQ 20$; YES - GO CALCULATE INDEX
437 000506 005316 15$: DEC (SP) ; DECREMENT COUNT - DONE LOOPING?
438 000510 001371 BNE 10$; NO
439 000512 000261 SEC ; YES - NO MATCH
440 000514 000404 BR 25$; RETURN WITH C-BIT SET
441
442 000516 010002 20$: MOV R0,R2 ; GET CURRENT POSITION IN PDV ADDRESS TABLE
443 000520 166702 000000G SUB $PDVTA,R2 ; SUBTRACT STARTING ADDRESS OF TABLE
444 000524 005742 TST -(R2) ; GET PROCESS PDV INDEX
445 000526 005226 25$: INC (SP)+ ; CLEAN UP STACK AND DON'T CHANGE C-BIT
446 000530 RESRG <R1,R0> ; RESTORE REGISTERS
447 000534 RETURN ; RETURN
448
;

```

58 000000  
59

; PDVDF\$

; DEFINE THE PDV OFFSETS

```

180
181 000210 $TK100:: ; NUMBER OF CLOCK TICKS IN 100 MSEC
182 000210 000006 $TK50:::WORD 6 ; NUMBER OF CLOCK TICKS IN 50 MSEC
183 000212 $SQRCM:::BYTE 0 ; SQ ROOT LIMIT FOR COMPONENT BUFFER ALLOCATION
184 000213 .BYTE 0 ; NUMBER OF ACTIVE COMPONENTS
185
186 000214 $MAXOV:::BLKW 1 ; MAXIMUM PROTOCOL OVERHEAD
187
188 000216 $LGPDV:::BLKW 1 ; PDV INDEX OF EVENT LOGGER
189 000220 $LGDDB:::BLKW 1 ; ADDRESS OF EVENT LOGGER DATABASE
190 000222 $FILHD:::BLKW 1 ; EVENT FILTER LISTHEAD
191 000224 $LGSTT:::BLKW 1 ; EVENT LOGGER STATUS
192
193 000226 $EVDSC:::BLKW 8. ; EVENT DESCRIPTOR BLOCK
194
195 000246 $LGFNB:::BLKW 3 ; EVENT LOGGER FILENAME BLOCK - FILE ID
196 000254 021265 055254 057330 .RAD50 /EVENTLOG/ ; FILENAME
197 000262 075273 .RAD50 /SYS/ ; FILE TYPE
198 000264 .BLKW ; VERSION NUMBER
199 000266 114 102 .ASCII /LB/ ; DEVICE NAME
200 000270 .BLKW ; AND UNIT
201 000272 006 001 $LGUIC:::BYTE 6,1 ; UIC FOR LOGGING FILE (DEFAULT = [1,6])
202 000274 051646 131574 $LGMON:::RAD50 /MON.../ ; MONITOR TASK NAME
203 000300 103 117 $LGCON:::ASCII /CO/ ; CONSOLE DEVICE NAME
204 000302 000000 .WORD 0 ; AND UNIT
205 000304 000C00 $CELFN:::WORD 0 ; CEX LOGGING FUNCTION CODE (MUST BE ZERO)
206
207 000306 000000 $NMCLH:::WORD 0 ; NETWORK MANAGEMENT COMPLETION LISTHEAD
208 000310 000306 $NMCL2:::WORD -2
209
210 .IF DF R$$$MPL
211 $NETPF:::WORD 0 ; POWERFAIL ENTRY POINT
212 $NTLAL:::WORD 0 ; NTL POINTER TO CEX POOL LISTHEAD
213
214 .IF DF M$$$PRO
215
216 TIMERS: .WORD -1 ; SHOW TIMER IS NOT ACTIVE
217 .WORD 0,0,0,0 ; PROCESSOR #1 DEPENDANT ONE SECOND
218 .WORD 0 ; TIMER QUEUE BLOCK
219 .WORD 0
220 .WORD 0
221 $$$STM=-TIMERS
222
223 .WORD 0,0,0,0 ; PROCESSOR #1 DEPENDANT 100 MSEC
224 .WORD 0 ; TIMER QUEUE BLOCK
225 .WORD 0
226 .WORD 0
227 $$$LEN=-TIMERS
228
229 .REPT M$$$PRO-1
230 .WORD -1 ; SHOW TIMER IS NOT ACTIVE
231 .WORD 0,0,0,0 ; PROCESSOR #2 - N DEPENDANT ONE SECOND
232 .WORD 0 ; TIMER QUEUE BLOCK
233 .WORD 0
234 .WORD 0
235
236 .WORD 0,0,0,0 ; PROCESSOR #2 - N DEPENDANT 100 MSEC

```

```
90 .SBTTL DEFINE KG-11 REGISTERS AND BITS
91
92 170700 KGCSR=170700 : KG-11 CSR
93 000001 CRC16=1 : USE CRC-16 POLYNOMIAL
94 000003 LRC16=3 : USE LRC-16 POLYNOMIAL
95 000010 DDB=10 : SET TO WORD MODE
96 000020 CLRG=20 : INITIALISE KG-11
97 000100 SEN=100 : ENTER CYCLE MODE
98 000133 KGLDBC=DDB:LRC16:CLRG:SEN : LOAD NEW BCC ACCUMULATION
99 000111 KGINIT=DDB:CRC16:SEN : INIT TO DO CRC-16 CALCULATION
```



|     |        |        |              |
|-----|--------|--------|--------------|
| 659 |        | .WORD  | 114401       |
| 660 |        | .WORD  | 54700        |
| 661 |        | .WORD  | 54200        |
| 662 |        | .WORD  | 114101       |
| 663 |        | .WORD  | 104001       |
| 664 |        | .WORD  | 44300        |
| 665 |        | .WORD  | 44600        |
| 666 |        | .WORD  | 104501       |
| 667 |        | .WORD  | 45400        |
| 668 |        | .WORD  | 105701       |
| 669 |        | .WORD  | 105201       |
| 670 |        | .WORD  | 45100        |
| 671 |        | .WORD  | 47000        |
| 672 |        | .WORD  | 107301       |
| 673 |        | .WORD  | 107601       |
| 674 |        | .WORD  | 47500        |
| 675 |        | .WORD  | 106401       |
| 676 |        | .WORD  | 46700        |
| 677 |        | .WORD  | 46200        |
| 678 |        | .WORD  | 106101       |
| 679 |        | .WORD  | 42000        |
| 680 |        | .WORD  | 102301       |
| 681 |        | .WORD  | 102601       |
| 682 |        | .WORD  | 42500        |
| 683 |        | .WORD  | 103401       |
| 684 |        | .WORD  | 43700        |
| 685 |        | .WORD  | 43200        |
| 686 |        | .WORD  | 103101       |
| 687 |        | .WORD  | 101001       |
| 688 |        | .WORD  | 41300        |
| 689 |        | .WORD  | 41600        |
| 690 |        | .WORD  | 101501       |
| 691 |        | .WORD  | 40400        |
| 692 |        | .WORD  | 100701       |
| 693 |        | .WORD  | 100201       |
| 694 |        | .WORD  | 40100        |
| 695 |        |        |              |
| 696 |        | .IFF   |              |
| 697 |        |        |              |
| 698 | 000316 | 000000 | .WORD 0      |
| 699 | 000320 | 146001 | .WORD 146001 |
| 700 | 000322 | 154001 | .WORD 154001 |
| 701 | 000324 | 012000 | .WORD 12000  |
| 702 | 000326 | 170001 | .WORD 170001 |
| 703 | 000330 | 036000 | .WORD 36000  |
| 704 | 000332 | 024000 | .WORD 24000  |
| 705 | 000334 | 162001 | .WORD 162001 |
| 706 | 000336 | 120001 | .WORD 120001 |
| 707 | 000340 | 066000 | .WORD 66000  |
| 708 | 000342 | 074000 | .WORD 74000  |
| 709 | 000344 | 132001 | .WORD 132001 |
| 710 | 000346 | 050000 | .WORD 50000  |
| 711 | 000350 | 116001 | .WORD 116001 |
| 712 | 000352 | 104001 | .WORD 104001 |
| 713 | 000354 | 042000 | .WORD 42000  |
| 714 |        |        |              |
| 715 |        | .ENDC  |              |

;SECOND HALF OF TABLE

STCRCF MACRO V05.03b Friday 28-Jun-85 18:22 Page 8  
EXECUTIVE VECTOR TABLE

```
101
102
103 000000 000000
104 000002 000000G
105 000004 172354
106
107 000002
```

```
.SBTTL EXECUTIVE VECTOR TABLE
$STCVT::WORD 0 ; FLAGS WORD
KSAR5: .WORD KISAR5 ;
KSAR6: .WORD KISAR6 ;
$STCVL == <<.-$STCVT>/2>-1 ; LENGTH OF VECTOR TABLE
```

STCRCF MACRO V05.03b Friday 28-Jun-85 18:22 Page 12-5  
MODIFIER TABLE FOR SOFTWARE CRC

N 7

716  
717  
718  
719

.ENDC

000001

.END

STCRCK MACRO V05.03b Friday 28-Jun-85 18:22 Page 9  
 \$CLCRC - CALCULATE CRC-16 ON A TRANSMIT CHAIN

```

109 .SBTTL $CLCRC - CALCULATE CRC-16 ON A TRANSMIT CHAIN
110
111 +
112 **-$CLCRC-CALCULATE CRC (CRC16) ON A TRANSMIT CHAIN
113 -
114 INPUTS:
115 R4 - ADDRESS OF FIRST CCB IN CHAIN
116 C.LNK - ADDRESS OF NEXT CCB IN CHAIN (0 MARKS END)
117 C.BUF - DOUBLE WORD ADDRESS OF BUFFER
118 C.CNT - LENGTH OF BUFFER
119 C.FLG - BUFFER FLAGS
120
121 OUTPUTS:
122 CRC IS STORED AT THE END OF EACH BUFFER AND THE BYTE
123 COUNT IS UPDATED.
124
125 REGISTERS MODIFIED:
126 R2, R3
127
128 000006 $CLCRC::
129
130 .IF DF M$$MGE
131 .IF NDF I$$AS
132 000006 017746 177770 MOV @KSAR5,-(SP) ; SAVE CALLER'S MAPPING
133
134 .IFF ; I$$AS
135
136 MOV #KP.AR3,-(SP) ; ASSUME KERNEL MODE
137 BIT #140000,PS.EXP ; IS IT ?
138 BEQ 5$; YES - BR
139 MOV #UPA:0+6,(SP) ; NO - MUST BE USER MODE
140 5$: MOV @ (SP),-(SP) ; SAVE CURRENT MAPPING
141
142 .ENDC ; I$$AS
143 .ENDC ; M$$MGE
144
145 000012 SAVRG <R0,R1,R4> ; SAVE REGISTERS
146 000020 005001 CLR R1 ; INITIALISE CRC ACCUMULATOR
147
148 000022 010403 10$: MOV R4,R3 ; COPY CCB ADDRESS
149 000024 062703 000014 ADD #C.BUF,R3 ; AND POINT TO BUFFER DESCRIPTOR
150
151 .IF DF M$$MGE
152 .IF NDF I$$AS
153
154 000030 012377 177746 MOV (R3)+,@KSAR5 ; MAP TO THE DATA
155
156 .IFF ; I$$AS
157
158 MOV (R3)+,@10(SP) ; MAP TO THE DATA
159
160 .ENDC ; I$$AS
161 .IFF ; M$$MGE
162
163 TST (R3)+ ; SKIP OVER RELOCATION BIAS
164
165 .IFTF ; M$$MGE

```

STCRCK MACRO V05.03b Friday 28-Jun-85 18:22 Page 12-6  
Symbol table

|                |                |                 |                  |                   |
|----------------|----------------|-----------------|------------------|-------------------|
| ASSCHK= 000000 | CS.ABO= 000100 | C.STS 000012    | FS.RNG= 011000   | M\$NET= 000000    |
| ASSCPS= 000000 | CS.BRO= 000002 | C.URM 177776    | FS.RST= 000000   | M\$OVR= 000000    |
| ASSPRI= 000000 | CS.BUF= 000200 | C.XACP 000004   | FS.RTN= 001000   | N\$ACC= 000001    |
| ASSTRP= 000000 | CS.CES= 000002 | C.XID 000035    | FS.SET= 005000   | N\$BUF= 000001    |
| BUFUMP= 172354 | CS.CHN= 000010 | C.XLEN 000044   | FS.SFC= 005000   | N\$LDV= 000001    |
| CB.CCB= 000002 | CS.CMP= 000200 | C.XPLI 000040   | FS.SFR= 006000   | N\$MCP= 000001    |
| CB.DDM= 000040 | CS.DCR= 000400 | C.XPT 000034    | FS.SFS= 004000   | N\$MLL= 000001    |
| CB.DLC= 000020 | CS.DEF= 000004 | C.XSVC 000042   | FS.SPW= 040000   | N\$MOV= 000010    |
| CB.RDB= 000004 | CS.DEV= 000002 | C.XTC 000037    | FS.STM= 000000   | N\$NCT= 000001    |
| CB.SDB= 000010 | CS.DIS= 000040 | C.X25 000036    | FS.STP= 002000   | N\$PEM= 000001    |
| CB.SLI= 000100 | CS.ENA= 000001 | DDB = 000010    | FS.STR= 001000   | PIRO = 177772     |
| CB.XLB= 000001 | CS.ENB= 000020 | D\$BUG= 177514  | FS.TRM= 003000   | PMODE = 030000    |
| CC.LLC= 000200 | CS.ERR= 100000 | D\$ISK= 000000  | FS.WLB= 001000   | PRO = 000000      |
| CE.ABO= 100362 | CS.FTL= 001000 | D\$LL1= 000001  | FS.XKL= 002000   | PR1 = 000040      |
| CE.DAO= 100346 | CS.HCR= 000001 | D\$SYNC= 000000 | FS.XGF= 010000   | PR2 = 000100      |
| CE.DIS= 100366 | CS.HFE= 002000 | D\$YNN= 000000  | FS.XON= 007000   | PR3 = 000140      |
| CE.ERR= 100370 | CS.LST= 040000 | E\$XPR= 000000  | FS.ZER= 002000   | PR4 = 000200      |
| CE.ILN= 100350 | CS.MTL= 004000 | FC.CCP= 000020  | F\$LVL= 000001   | PR5 = 000240      |
| CE.LTO= 100356 | CS.RNG= 000010 | FC.CTL= 000006  | G\$TPP= 000000   | PR6 = 000300      |
| CE.MOP= 100372 | CS.ROV= 000004 | FC.KCP= 000016  | G\$TSS= 000000   | PR7 = 000340      |
| CE.NTE= 100361 | CS.RSN= 010000 | FC.KIL= 000004  | G\$TTK= 000000   | PS = 177776       |
| CE.RTE= 100376 | CS.SHU= 000001 | FC.MAN= 000024  | G\$WRD= 000000   | P\$P45= 000000    |
| CE.SRC= 100364 | CS.SID= 000002 | FC.MLD= 000026  | I\$RAR= 000000   | P\$WRD= 000000    |
| CE.STP= 100352 | CS.STP= 000004 | FC.PCT= 000030  | I\$RDN= 000000   | Q\$OPT= 000010    |
| CE.TME= 100354 | CS.SUC= 000001 | FC.PWR= 000022  | KGCSR = 170700   | R\$DER= 000000    |
| CE.TMO= 100374 | CS.TMO= 020000 | FC.RCE= 000002  | KGINIT= 000111   | R\$EIS= 000000    |
| CE.UNS= 100344 | CS.XUR= 000004 | FC.RCP= 000014  | KGLDBC= 000133   | R\$K11= 000001    |
| CF.CHN= 000001 | C\$CKP= 000000 | FC.TJM= 000010  | KISAR0= 172340   | R\$SND= 000000    |
| CF.EOM= 000004 | C\$ORE= 000400 | FC.XCP= 000012  | KISAR5= ***** Gx | R\$11M= 000000    |
| CF.HDR= 000020 | C\$RSH= 177564 | FC.XME= 000000  | KISAR6= 172354   | SEN = 000100      |
| CF.LB = 100000 | C.ADD 000034   | FS.AST= 000000  | KSAR5 000002R    | STCR2 000162R     |
| CF.LIN= 000002 | C.BID 000003   | FS.CIB= 002000  | KSAR6 000004R    | SWR = 177570      |
| CF.SOM= 000010 | C.BUF 000014   | FS.CRA= 001000  | K\$CNT= 177546   | S\$WRG= 000000    |
| CF.SYN= 000040 | C.BUF1 000014  | FS.DIS= 013000  | K\$CSR= 177546   | S\$YSZ= 007600    |
| CF.TRN= 000100 | C.BUF2 000024  | FS.DVC= 001000  | K\$G11= 000001   | TPS = 177564      |
| CLRKG = 000020 | C.CNT 000020   | FS.ENB= 012000  | K\$LDC= 000000   | T\$KMG= 000000    |
| CMODE = 140000 | C.CNT1 000020  | FS.EXI= 001000  | K\$TPS= 000074   | T\$MIN= 000000    |
| CM.CIR= 000002 | C.CNT2 000030  | FS.GET= 006000  | L\$LP = 000000   | UBMPR = 170200    |
| CM.FMT= 100000 | C.FLG 000022   | FS.HLT= 000000  | LRC16 = 000003   | UISAR0= 177640    |
| CM.HRD= 000002 | C.FLG1 000022  | S.INI= 000000   | L\$ASG= 000000   | UISAR1= 177642    |
| CM.LIN= 000000 | C.FLG2 000032  | FS.KIL= 000000  | L\$DRV= 000000   | V\$CTR= 001000    |
| CM.LOD= 000001 | C.FNC 000010   | FS.LCL= 100000  | L\$P11= 000001   | X\$DBT= 000000    |
| CM.XLO= 000004 | C.LIN 000006   | FS.LTM= 001000  | L\$11R= 000000   | \$CLCRC 000006RG  |
| CP.DCF= 000040 | C.LNK 000000   | FS.MNT= 004000  | MPAR = 172100    | \$STCRC 000122RG  |
| CP.HDL= 000007 | C.MOD 000011   | FS.MSN= 014000  | MPLSR = 177746   | \$STCR1 000124RG  |
| CP.PS = 177400 | C.NSP 000004   | FS.REA= 001000  | M\$CRB= 000124   | \$STCVL= 000002 G |
| CP.PSI= 000200 | C.PRO 000042   | FS.RET= 000000  | M\$CRX= 000000   | \$STCVT 000000RG  |
| CP.XCF= 000100 | C.RSV 000002   | FS.REZ= 003000  | M\$FCS= 000000   | .BASEB= 140000    |
| CP.2FR= 000030 | C.STA 000007   | FS.RLB= 002000  | M\$MGE= 000000   | .\$\$\$= 000034   |
| CRC16 = 000001 |                |                 |                  |                   |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000336 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

AXBFR MACRO V05.03b Friday 28-Jun-85 18:28 Page 6-3  
Symbol table

|                |                 |                 |                |                  |
|----------------|-----------------|-----------------|----------------|------------------|
| ASSCHK= 000000 | CS.DIS= 000040  | FC.CTL= 000006  | KSSCNT= 177546 | MSSCPX= 000000   |
| ASSCPS= 000000 | CS.ENA= 000001  | FC.KCP= 000016  | KSSCSR= 177546 | MSSFLS= 000000   |
| ASSPRI= 000000 | CS.ENB= 000020  | FC.KIL= 000004  | KSSLDC= 000000 | MSSMGE= 000000   |
| ASTRP= 000000  | CS.ERR= 100000  | FC.MAN= 000024  | KSSTPS= 000074 | MSSNET= 000000   |
| BUFUMP= 172354 | CS.FTL= 001000  | FC.MLD= 000026  | LDLSP= 000000  | MSSOVR= 000000   |
| CB.CCH= 000002 | CS.HCR= 000001  | FC.PCT= 000030  | LF.ACT= 100000 | MSSACC= 000001   |
| CB.DDM= 000040 | CS.HFE= 002000  | FC.PWR= 000022  | LF.BRO= 000400 | MSSBUF= 000001   |
| CB.DLC= 000020 | CS.LST= 040000  | FC.RCE= 000002  | LF.BWT= 000007 | MSSLDV= 000001   |
| CB.RDB= 000004 | CS.MTL= 004000  | FC.RCP= 000014  | LF.ENA= 002000 | MSSMCP= 000001   |
| CB.SDB= 000010 | CS.RNG= 000010  | FC.TJM= 000010  | LF.LPB= 001000 | MSSMLL= 000001   |
| CB.SLI= 000100 | CS.ROV= 000004  | FC.XCP= 000012  | LF.MDC= 000100 | MSSMOV= 000010   |
| CB.XLB= 000001 | CS.RSN= 010000  | FC.XME= 000000  | LF.MFL= 004000 | MSSNCT= 000001   |
| CC.LLC= 000200 | CS.SHU= 000001  | FS.AST= 000000  | LF.MTP= 000020 | MSSPEM= 000001   |
| CE.ABO= 100362 | CS.SID= 000002  | FS.CIB= 002000  | LF.PAC= 000200 | PDSPPL= ***** GX |
| CE.DAO= 100346 | CS.STR= 000004  | FS.CRA= 001000  | LF.RDY= 040000 | PIRQ = 177772    |
| CE.DIS= 100366 | CS.SUC= 000001  | FS.CIS= 013000  | LF.REA= 010000 | PMODE = 030000   |
| CE.ERR= 100370 | CS.TMO= 020000  | FS.DVC= 001000  | LF.SER= 000040 | PR0 = 000000     |
| CE.ILN= 100350 | CS.XUR= 000004  | FS.ENB= 012000  | LF.TJM= 000010 | PR1 = 000040     |
| CE.LTO= 100356 | CS.CKP= 000000  | FS.EXI= 001000  | LF.UNL= 020000 | PR2 = 000100     |
| CE.MOP= 100372 | CS.SRE= 000400  | FS.GET= 006000  | LF.X2P= 000000 | PR3 = 000140     |
| CE.NTE= 100361 | CS.SRSH= 177564 | FS.HLT= 000000  | LN.CLO= 000000 | PR4 = 000200     |
| CE.RTE= 100376 | C.ADD= 000034   | FS.INJ= 000000  | LN.DUM= 000005 | PR5 = 000240     |
| CE.SRC= 100364 | C.BID= 000003   | FS.KIL= 000000  | LN.LOA= 000004 | PR6 = 000300     |
| CE.STP= 100352 | C.BUF= 000014   | FS.LCL= 100000  | LN.LOO= 000003 | PR7 = 000340     |
| CE.TMC= 100354 | C.BUF1= 000014  | FS.LTM= 001000  | LN.OAU= 000003 | PS = 177776      |
| CE.TMO= 100374 | C.BUF2= 000024  | FS.MNT= 004000  | LN.OFF= 000001 | PSSP45= 000000   |
| CE.UNS= 100344 | C.CNT= 000020   | FS.MSN= 014000  | LN.ON= 000000  | PSSWRD= 000000   |
| CF.CHN= 000001 | C.CNT1= 000020  | FS.REA= 001000  | LN.OOP= 000004 | QSSOPT= 000010   |
| CF.EOM= 000004 | C.CNT2= 000030  | FS.REI= 000000  | LN.OPE= 000001 | RDBG1 = ***** GX |
| CF.HDR= 000020 | C.FLG= 000022   | FS.REZ= 003000  | LN.REF= 000002 | RDBRT = ***** GX |
| CF.LB= 100000  | C.FLG1= 000022  | FS.RLB= 002000  | LN.SER= 000002 | RDBSZ = ***** GX |
| CF.LIN= 000002 | C.FLG2= 000032  | FS.RNG= 011000  | LN.STA= 000017 | RDQCT = ***** GX |
| CF.SOM= 000010 | C.FNC= 000010   | FS.RST= 000000  | LN.SUB= 000360 | RDQSL = ***** GX |
| CF.SYN= 000040 | C.LIN= 000006   | FS.RTN= 001000  | LN.TRI= 000006 | RSSDER= 000000   |
| CF.TRN= 000100 | C.LNK= 000000   | FS.SET= 005000  | L\$ASG= 000000 | RSSK11= 000001   |
| CMODE= 140000  | C.MOD= 000011   | FS.SFC= 005000  | L\$DRV= 000000 | RSSND= 000000    |
| CM.CIR= 000002 | C.NSP= 000004   | FS.SFR= 006000  | L\$P11= 000001 | RSS11M= 000000   |
| CM.FMT= 100000 | C.PRO= 000042   | FS.SFS= 004000  | L\$T1R= 000000 | SF.ACT= 000200   |
| CM.HRD= 000002 | C.RSV= 000002   | FS.SPW= 040000  | L.COST= 000015 | SF.ENA= 000100   |
| CM.LIN= 000000 | C.STA= 000007   | FS.STM= 000000  | L.CTL= 000012  | SF.LPB= 000004   |
| CM.LOO= 000001 | C.STS= 000012   | FS.SIP= 002000  | L.CVA= 177776  | SF.MFL= 000040   |
| CM.XLO= 000004 | C.URM= 177776   | FS.STR= 001000  | L.DDM= 000002  | SF.PAC= 000020   |
| CP.DCF= 000040 | C.XACP= 000004  | FS.TRM= 003000  | L.DDS= 000004  | SF.REA= 000010   |
| CP.HDL= 000007 | C.XID= 000035   | FS.WLB= 001000  | L.DLC= 000003  | SF.SER= 000001   |
| CP.PS= 177400  | C.XLEN= 000044  | FS.XKL= 002000  | L.DLM= 000006  | SF.SVC= 000002   |
| CP.PSI= 000200 | C.XPLI= 000040  | FS.XOF= 010000  | L.DLS= 000010  | SF.UNL= 000040   |
| CP.XCF= 000100 | C.XPT= 000034   | FS.XON= 007000  | L.FLG= 000000  | SLTMA = ***** GX |
| CP.2FR= 000030 | C.XSVC= 000042  | FS.ZER= 002000  | L.KRBA= 000016 | SLTNM = ***** GX |
| CS.ABO= 000100 | C.XTC= 000037   | F\$SLVL= 000031 | L.LEN= 000022  | SWR = 177570     |
| CS.BRO= 000002 | C.X25= 000036   | G\$STPP= 000000 | L.MPF= 000022  | S\$SWRG= 000000  |
| CS.BUF= 000200 | D\$BUG= 177514  | G\$STSS= 000000 | L.NMST= 000020 | S\$YSZ= 007600   |
| CS.CES= 000002 | D\$ISK= 000000  | G\$STTK= 000000 | L.NSTA= 000014 | S.COST= 000001   |
| CS.CHN= 000010 | D\$LL1= 000001  | G\$SWRD= 000000 | L.OWNR= 000021 | S.FLG= 000000    |
| CS.CMP= 000200 | D\$SYNC= 000000 | I\$SRAR= 000000 | L.UNT= 000013  | S.LEN= 000004    |
| CS.DCR= 000400 | D\$SYNM= 000000 | I\$SRDN= 000000 | MPAR = 172100  | S.NMST= 000002   |
| CS.DEF= 000004 | E\$XPR= 000000  | KISARO= 172340  | MPCR= 177746   | S.OWNR= 000003   |
| CS.DEV= 000002 | FC.CCP= 000020  | KISAR6= 172354  | M\$CRB= 000124 | TPS = 177564     |

AXDAT CREATED BY MACRO ON 28-JUN-85 AT 18:28

PAGE 2 N 11

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES |
|---------|------------|------------|
| T1SCL   | 000146 RG  | #5-140     |
| T100C   | 000150 RG  | #5-141     |
| T100Q   | 000152 RG  | #5-142     |
| UMRPT   | 000030 RG  | #5-76      |
| XAVL    | 000154 RG  | #5-143     |
| ZTIME   | 000156 RG  | #5-144     |
| ZTIM2   | 000160 RG  | #5-145     |
| \$ALOCB | = ***** GX | 5-50       |
| \$ASCMP | = ***** GX | 5-86       |
| \$AUXVL | = 000070 G | #5-164     |
| \$AUXVT | 000000 RG  | #5-45      |
| \$CCBGT | = ***** GX | 5-P7       |
| \$CCBRT | = ***** GX | 5-88       |
| \$CEACC | = ***** GX | 5-89       |
| \$CEDIV | = ***** GX | 5-90       |
| \$CLINS | = ***** GX | 5-52       |
| \$CMFRK | = ***** GX | 5-91       |
| \$CXOPT | = ***** GX | 5-92       |
| \$DDAST | = ***** GX | 5-93       |
| \$DDCCP | = ***** GX | 5-94       |
| \$DDFNC | = ***** GX | 5-95       |
| \$DDMSN | = ***** GX | 5-96       |
| \$DEACB | = ***** GX | 5-60       |
| \$DSPTM | = ***** GX | 5-97       |
| \$EXRQF | = ***** GX | 5-61       |
| \$EXRON | = ***** GX | 5-62       |
| \$FMASK | = ***** GX | 5-70       |
| \$INTCT | = ***** GX | 5-72       |
| \$NMCLH | = ***** GX | 5-105      |
| \$NMCL2 | = ***** GX | 5-109      |
| \$PDDSP | = ***** GX | 5-111      |
| \$PDQUE | = ***** GX | 5-112      |
| \$PDSPL | = ***** GX | 5-113      |
| \$PDVNM | = ***** GX | 5-114      |
| \$PDVTA | = ***** GX | 5-115      |
| \$PUMR  | = ***** GX | 5-116      |
| \$PWRF1 | = ***** GX | 5-117      |
| \$QSTRT | = ***** GX | 5-118      |
| \$RDBG1 | = ***** GX | 5-119      |
| \$RDBNM | = ***** GX | 5-120      |
| \$RDBRT | = ***** GX | 5-121      |
| \$RDBSZ | = ***** GX | 5-122      |
| \$RDBTH | = ***** GX | 5-123      |
| \$RDQCT | = ***** GX | 5-124      |
| \$RDQSL | = ***** GX | 5-125      |
| \$SLTMA | = ***** GX | 5-126      |
| \$SLTNM | = ***** GX | 5-127      |
| \$SORCM | = ***** GX | 5-128      |
| \$SRSTD | = ***** GX | 5-73       |
| \$STD01 | = ***** GX | 5-129      |
| \$STDLC | = ***** GX | 5-130      |
| \$STDLT | = ***** GX | 5-131      |

AXDSP - AUXILLIARY PROCESS DISP MACRO V05.03b Tuesday 03-Sep-85 <sup>N 12</sup> 10:58 Page 9-1  
NETWORK MANAGEMENT COUNTER COMPLETION

311 000236  
312 000240 001373  
313 000242  
314  
315 000001

RESRG <R4>  
BNE 70\$  
RETURN  
.  
END

; Recover the next buffer  
; If NE, get next buffer  
; Else, return



AXDSPB - AUXILLIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 <sup>N 13</sup> 10:28 Page 8-1  
POWERFAIL RECOVERY ROUTINE

179 000164

CALLR MDMSCN ; MODEM CONTROL SCAN ROUTINE

180

.IFF

181

182

RETURN

183

184

.ENDC

185

186

187

;+ LOCAL STORAGE FOR TRIBUTARY ADDRESS DURING POWERFAIL  
;-

188

189

190

191 000170

TRIB: .BLKW

|    |   |                                                                   |
|----|---|-------------------------------------------------------------------|
| 58 | : | DECNET-11M/S V3.0                                                 |
| 59 | : | DECNET-11M-PLUS V1.0                                              |
| 60 | : |                                                                   |
| 61 | : | 4.00 07-NOV-83                                                    |
| 62 | : | DECNET-11M V4.0                                                   |
| 63 | : | DECNET-11M-PLUS V2.0                                              |
| 64 | : |                                                                   |
| 65 | : | 5.00 22-JUL-85                                                    |
| 66 | : | DECnet-11M/S V4.2                                                 |
| 67 | : | DECnet-11M-Plus V3.0                                              |
| 68 | : | DECnet-Micro/R SX V1.0                                            |
| 69 | : |                                                                   |
| 70 | : | 5.01 09-Aug-85                                                    |
| 71 | : | Add logic to bypass service disabled check on broadcast channels. |
| 72 | : | This change requires EPMMAI ident V5.01.                          |
| 73 | : |                                                                   |

AXDSPM      CREATED BY    MACRO    ON 3-SEP-85 AT 10:59

PAGE 3      N 15

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|         |        |        |        |
|---------|--------|--------|--------|
| CALL    | 10-294 | 10-310 |        |
| CALLR   | 7-120  | 10-304 | 10-308 |
| CCBDF\$ | #6-75  | 6-78   |        |
| RESRG   | #6-75  | 10-311 |        |
| RETURN  | 7-119  | 10-313 |        |
| SAVRG   | #6-75  | 10-309 |        |
| SLTDF\$ | #6-75  | 6-77   |        |
| SOB     | 10-289 |        |        |

[illegible]

|   |    |        |
|---|----|--------|
| J | 5  |        |
| K | 5  |        |
| L | 5  |        |
| M | 5  |        |
| N | 5  |        |
| B | 6  |        |
| C | 6  |        |
| D | 6  |        |
| E | 6  |        |
| F | 6  |        |
| G | 6  | STCRCF |
| H | 6  |        |
| I | 6  |        |
| J | 6  |        |
| K | 6  |        |
| L | 6  |        |
| M | 6  |        |
| N | 6  |        |
| B | 7  |        |
| C | 7  |        |
| D | 7  |        |
| E | 7  |        |
| F | 7  |        |
| G | 7  |        |
| H | 7  |        |
| I | 7  |        |
| J | 7  |        |
| K | 7  |        |
| L | 7  |        |
| M | 7  |        |
| N | 7  |        |
| B | 8  |        |
| C | 8  |        |
| D | 8  |        |
| E | 8  |        |
| F | 8  | STCRCK |
| G | 8  |        |
| H | 8  |        |
| I | 8  |        |
| J | 8  |        |
| K | 8  |        |
| L | 8  |        |
| M | 8  |        |
| N | 8  |        |
| B | 9  |        |
| C | 9  |        |
| D | 9  |        |
| E | 9  |        |
| F | 9  |        |
| G | 9  |        |
| H | 9  |        |
| I | 9  |        |
| J | 9  |        |
| K | 9  |        |
| L | 9  |        |
| M | 9  |        |
| N | 9  |        |
| B | 10 |        |
| C | 10 |        |
| D | 10 |        |

| AXBFR | AXDAT | AXDSP | AXDSPB | AXDSPM |
|-------|-------|-------|--------|--------|
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |
| 4     |       |       |        |        |
| 3     |       |       |        |        |
| 2     |       |       |        |        |
| 1     |       |       |        |        |
| 0     |       |       |        |        |
| 10    |       |       |        |        |
| 9     |       |       |        |        |
| 8     |       |       |        |        |
| 7     |       |       |        |        |
| 6     |       |       |        |        |
| 5     |       |       |        |        |

|   |    |
|---|----|
| M | 14 |
| N | 14 |
| B | 15 |
| C | 15 |
| D | 15 |
| E | 15 |
| F | 15 |
| G | 15 |
| H | 15 |
| I | 15 |
| J | 15 |
| K | 15 |
| L | 15 |
| M | 15 |
| N | 16 |
| B | 16 |
| C | 16 |
| D | 16 |
| E | 16 |
| F | 16 |
| G | 16 |
| H | 16 |
| I | 16 |
| J | 16 |
| K | 16 |
| L | 16 |
| M | 16 |

```

Symbol table
ASCOMP = *****
ASSCHK= 000000
ASSCPS= 000000
ASSPRI= 000000
ASSTRP= 000000
CB.CCB= 000002
CB.DDM= 000040
CB.DLC= 000020
CB.RDB= 000004
CB.SDB= 000010
CB.SLI= 000100
CB.XLB= 000001
CCBRET= 000030R
CCBRT = ***** GX
CC.LLC= 000200
CE.ABO= 100362
CE.DAO= 100346
CE.DIS= 100366
CE.ERR= 100370
CE.ILN= 100350
CE.LTO= 100356
CE.MOP= 100372
CE.NTE= 100361
CE.RTE= 100376
CE.SRC= 100364
CE.STP= 100352
CE.TME= 100354
CE.TMO= 100374
CE.UNS= 100344
CF.CHN= 000001
CF.EQM= 000004
CF.HDR= 000020
CF.LB = 100000
CF.LIN= 000002
CF.SOM= 000010
CF.SYN= 000040
CF.TRN= 000100
CM.CIR= 000002
CM.FMT= 100000
CM.HRD= 000002
CM.LIN= 000000
CM.LOD= 000001
CM.XLO= 000004
CP.DCF= 000040
CP.HDL= 000007
CP.PS = 177400
CP.PSI= 000200
CP.XCF= 000100
CP.2FR= 000030
CS.ABO= 000100
CS.BRC= 000002
CS.BUF= 000200
CS.CES= 000002
CS.CHN= 000100
CS.CMP= 000200
CS.DCR= 000400
CS.DEF= 000004
CS.DEV= 000002
CS.DIS= 000040
CS.ENA= 000001
CS.ENB= 000020
CS.ERR= 100000
CS.FTL= 001000
CS.HCR= 000001
CS.HFE= 002000
CS.LST= 040000
CS.MTL= 004000
CS.RNG= 000010
CS.ROV= 000004
CS.RSN= 010000
CS.SHU= 000001
CS.SID= 000002
CS.STR= 000004
CS.SUC= 000001
CS.TMO= 020000
CS.XUR= 000004
CSSCKP= 000000
CSSORE= 000400
CSSRSH= 177564
C.ADD 000034
C.BID 000003
C.BUF 000014
C.BUF1 000014
C.BUF2 000024
C.CNT 000020
C.CNT1 000020
C.CNT2 000030
C.FLG 000022
C.FLG1 000022
C.FLG2 000032
C.FNC 000010
C.LIN 000006
C.LNK 000000
C.MOD 000011
C.NSP 000004
C.PRO 000042
C.RSV 000002
C.STA 000007
C.STS 000012
C.URM 177776
C.XACP 000004
C.XID 000035
C.XLEN 000044
C.XPLI 000040
C.XPT 000034
C.XSVC 000042
C.XTC 000037
C.X25 000036
DDFNC = ***** GX
DUMMY= 000026R
DSSBUG= 177514
DSSISK= 000000
DSSL11= 000001
DSSYNC= 000000
DSSYNM= 000000
EXRON = ***** GX
ESSXPR= 000000
FC.CCP= 000020
FC.CTL= 000006
FC.KCP= 000016
FC.KIL= 000004
FC.MAN= 000024
FC.MLD= 000026
FC.PCT= 000030
FC.PWR= 000022
FC.RCE= 000002
FC.RCP= 000014
FC.TIM= 000010
FC.XCP= 000012
FC.XME= 000000
FS.AST= 000000
FS.CIB= 002000
FS.CRA= 001000
FS.DIS= 013000
FS.DVC= 001000
FS.ENB= 012000
FS.EXI= 001000
FS.GET= 006000
FS.HLT= 000000
FS.INI= 000000
FS.KIL= 000000
FS.LCL= 100000
FS.LTM= 001000
FS.MNT= 004000
FS.MSN= 014000
FS.REA= 001000
FS.RET= 000000
FS.REZ= 003000
FS.RLB= 002000
FS.RNG= 011000
FS.RST= 000000
FS.RTN= 001000
FS.SET= 005000
FS.SFC= 005000
FS.SFR= 006000
FS.SFS= 004000
FS.SPW= 040000
FS.STM= 000000
FS.STP= 002000
FS.STR= 001000
FS.TRM= 003000
FS.WLB= 001000
FS.XKL= 002000
FS.XOF= 010000
FS.XON= 007000
FS.ZER= 002000
FSLVL= 000001
GSSTPP= 000000
GSSITS= 000000
GSSITK= 000000
GSSWRD= 000000
ISSRAR= 000000
ISSRDN= 000000
KMCL 000300R
KSSCNT= 177546
KSSCSR= 177546
KSSLDC= 000000
KSSTPS= 000074
LDLPL = 000000
LF.ACT= 100000
LF.BRO= 000400
LF.BWT= 000007
LF.ENA= 002000
LF.LPB= 001000
LF.MDC= 000100
LF.MFL= 004000
LF.MTP= 000020
LF.PAC= 000200
LF.RDY= 040000
LF.REA= 010000
LF.SER= 000040
LF.TIM= 000010
LF.UNL= 020000
LF.X2P= 000000
LN.CLO= 000000
LN.DUM= 000005
LN.LOA= 000004
LN.LOD= 000003
LN.OAU= 000003
LN.OFF= 000001
LN.ON = 000000
LN.OOP= 000004
LN.OPE= 000001
LN.REF= 000002
LN.SER= 000002
LN.STA= 000017
LN.SUB= 000360
LN.TRI= 000006
LSSASG= 000000
LSSDRV= 000000
LSSP11= 000001
LSS11R= 000000
L.COST 000015
L.CTL 000012
L.CVA 177776
L.DDM 000002
L.DDS 000004
L.DLC 000003
L.DLM 000006
L.DLS 000010
L.FLG 000000
L.KRBA 000016
L.LEN = 000022
L.MPF 000022
L.NMST 000020
L.NSTA 000014
L.OWNR 000021
L.UNT 000013
MSSCRB= 000124
MSSCRX= 000000
MSSFC= 000000
MSSMGE= 000000
MSSNET= 000000
MSSOVR= 000000
NETACP 000304R
NMCL2 = ***** GX
NMCMP 000310R
NSSACC= 000001
NSSBUF= 000001
NSSLDV= 000001
NSSMCP= 000001
NSSMLL= 000001
NSSMOV= 000010
NSSNCT= 000001
NSSPEM= 000001
PDSP1 = ***** GX
PWFAIL 00034R
PWRFL 000170R
PWRF1 = ***** GX
PSSP45= 000000
PSSRFL= 000001
PSSWRD= 000000
QSSOPT= 000010
RDBRT = ***** GX
RSSDER= 000000
RSSK11= 000001
RSSND= 000000
RSS11M= 000000
SF.ACT= 000200
SF.ENA= 000100
SF.LPB= 000004
SF.MFL= 000040
SF.PAC= 000020
SF.REA= 000010
SF.SER= 000001
SF.SVC= 000002
SF.UNL= 000040
SLTMA = ***** GX
SLTMM = ***** GX
SRSTD = ***** GX
SSSWRG= 000000
SSSYST= 007600
S.COST 000001
S.FLG 000000
S.LEN 000004
S.NMST 000002
S.OWNR 000003
TRIB 000166R
TSKRT = ***** GX
TSSKMG= 000000
TSSMIN= 000000
VSSCTR= 001000
XSSDBT= 000000
$AUXTB 000000RG
$BFRN= ***** GX

```

```

222 .SBTTL D'SABLE LINE
223
224 ;+
225 ;**~DSABLE~DISABLE LINE
226
227 ;DISABLE THE LINE BY ENTERING A STATE WHICH WILL EVENTUALLY LEAD TO
228 ;IDLE. IF WE ARE WAITING FOR A CONNECT TO COMPLETE, POST A
229 ;COMPLETION ON THE ENABLE WITH AN ERROR STATUS.
230
231 ;-
232 INPUTS:
233 R5 - POINTER TO LINE ENTRY IN MDC DATABASE
234
235 OUTPUTS:
236 ONLY R4 WILL BE GUARANTEED TO BE PRESERVED
237
238 DSABLE: MOVB #TM.DIS, M.TIM(R5) ; START DISCONNECT TIMER
239 CMPB M.STT(R5), #ST.WCN ; IF WE ARE WAITING FOR A CONNECT
240 BEQ 10$; THEN GOTO POST ABORT TO DLC LEVEL
241 CMPB M.STT(R5), #ST.CDL ; IF WAITING FOR AN ASYNC CONNECT
242 BEQ 10$; THEN GOTO POST ABORT
243 .IF DF X$D52
244 CMPB M.STT(R5), #ST.DLY ; If waiting for CD and CTS
245 BEQ 10$; then Goto post abort
246 .ENDC
247 MOVB #ST.DDL, M.STT(R5) ; ELSE SET STATE TO DISCONNECT DELAY
248 RETURN
249
250 10$: CALL $PABO ; POST ABORT TO DLC LEVEL
251 BCC 20$; ALLOCATION FAILURE?
252
253 ;+
254 ;A CCB ALLOCATION FAILURE HAS OCCURRED. ENTER POST ABORT NOTIFICATION
255 ;STATE AND FORCE A TIMEOUT.
256 ;-
257 MOVB #ST.ABO, M.STT(R5) ; SET STATE TO POST ABORT NOTIFICATION
258 BISB #MC.CCB, M.PSV(R5) ; FORCE ^ DISPATCH ON NEXT TIMEOUT
259 20$: RETURN

```

```

610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625 001006 132765 000200 000005 .CONE:
626 001014 001410
627 001016
628

 .SBTTL ST.CER - CONNECT ERROR
 *--.CONE-CONNECT ERROR
 NOTIFY THE DLC THAT THE LINE HAS LOST CARRIER (ASYNC) OR DATASET
 READY (SYNC) IF IT HAS NOT ALREADY BEEN NOTIFIED. WAIT UNTIL
 THE DLC ISSUES A DISABLE.
 INPUTS:
 R2 - POINTER TO SYSTEM LINE TABLE
 R4 - SYSTEM LINE NUMBER
 R5 - POINTER TO LINE ENTRY IN MDC DATABASE
 .ENABL LSB
 .CONE: BITB #MC.CCB,M.PSV(R5) ; HAVE WE ALREADY NOTIFIED THE DLC?
 BEQ 10$; YES
 CALLR DISC ; POST ASYNCHRONOUS COMPLETION TO DLC

```

1 .TITLE AXSCH  
2 .IDENT /V05.00/  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51

3 COPYRIGHT (C) 1979,1980, 1982, 1983, 1985 BY  
4 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

5 THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
6 ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
7 INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
8 COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
9 OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
10 TRANSFERRED.

11 THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
12 AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
13 CORPORATION.

14 DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
15 SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

16 MODULE DESCRIPTION

17 CEX PROCESS SCHEDULING ROUTINES (RESIDENT IN AUX)

18 DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

19 IDENT HISTORY:

20 1.00 14-DEC-79  
21 DECNET-11M/S V3.0  
22 DECNET-11M-PLUS V1.0

23 3.00 16-APR-82  
24 DECNET-11M V3.1  
25 DECNET-11M-PLUS V1.1

26 4.00 07-NOV-83  
27 DECNET-11M V4.0  
28 DECNET-11M-PLUS V2.0

29 5.00 22-JUL-85  
30 DECnet-11M/S V4.2  
31 DECnet-11M-Plus V3.0  
32 DECnet-Micro/RSX V1.0



```

51 .SBTTL Macro definitions
52
53 ;
54 ; Macro Library Calls
55 ;
56 .MCALL INHIB$, ENABL$, SAVRG, RESRG
57 .MCALL CCBDF$, PDVDF$
58 .MCALL SAVMAP, RESMAP, MAP
59 .MCALL CALLR ; Avoid system dependency
60
61 CCBDF$; Define the ccb offsets
62 PDVDF$; Define the pdv offsets
63
64 ;
65 ; Local Storage
66 ;
67 .IF DF M$$MGE
68 000000 000003
69 000002
70 000004 000000
71
72 .ENDC
73
74 ;
75 ; Square root lookup table (uses scaling factor=2)
76 ; number = 2*(square root of components)
77
78 $SQRTB: .BYTE 2 ; 1 component running
79 .BYTE 3 ; 2 components running
80 .BYTE 3 ; 3 components running
81 .BYTE 4 ; 4 components running
82 .BYTE 4 ; 5 components running
83 .BYTE 5 ; 6 components running
84
85 ;
86 ;
87 ; Microcode Load Request Data
88 ;
89
90 000014 051444 131574
91 000006
92
93 MLDTSK: .RAD50 /MLD.../ ; RAD50 name of microcode loader
94 MAXLEN = 6 ; Maximum length of microcode name

```

AXSUB MACRO V05.03b Friday 28-Jun-85 18:30 Page 12-3  
Symbol table

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 15813 Words ( 62 Pages)  
Size of core pool: 16552 Words ( 63 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:17.11

SY:AXSUB.V2,[131,134]AXSUB/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]AXSUB

```

280 .SBTTL 100 MSEC TIMER SERVICE
281 :+
282 :*-PR100M-100 MSEC TIMER SERVICE
283 :
284 : THIS ROUTINE IS CALLED EVERY 100 MSEC (WHEN ACTIVE) FROM THE FORK
285 : LEVEL. IT WILL DISPATCH TIMEOUTS TO THOSE PROCESSES REQUESTING
286 : SHORT TIMER SERVICE WITH A SUBFUNCTION CODE OF 'FS.STM'.
287 :-
288 :
289 .ENABL LSB
290
291 000244 016700 000000G PR100M::MOV T100Q,R0 ; POINT TO SHORT TIMER QUEUE LIST HEAD
292 000250 CALL SCANSF ; SCAN THE SHORT TIMER QUEUE
293
294 000254 005777 000000G $T100I: TST @T100Q ; ANY ENTRIES LEFT IN THE QUEUE?
295 000260 001406 BEQ 10$; IF EQ, NO
296
297 000262 016700 000000G MOV T100C,R0 ; POINT TO THE 100 MSEC CLOCK BLOCK
298 000266 017702 000000G MOV @TK100,R2 ; GET # OF TICKS IN 100 MSEC
299 000272 CALL CLCINS ; AND INSERT ENTRY INTO CLOCK QUEUE
300
301 000276 10$: RETURN

```

SESCON - Session control connec MACRO V05.03b Friday 28-Jun-85 19:53<sup>B 8</sup>  
Table of contents

```

6- 42 Macro definitions
7- 63 Connect and connect accept QIO processing
8- 113 Connect QIO processing
9- 169 Connect accept QIO processing
10- 204 Process incoming connect request
11- 333 Queue connect request to task

```

SESCON - Session control connect MACRO V<sup>8</sup> 03b Friday 28-Jun-85 19:53 Page 11-4

# Symbol table

|                |               |                   |                   |                   |
|----------------|---------------|-------------------|-------------------|-------------------|
| WK.INT= 000020 | W.CSND 000020 | W.SEGZ 000006     | \$CNQIO 000000RG  | \$SRDSC= ***** GX |
| WK.RCV= 000004 | W.CTL 000000  | W.SNDQ 000016     | \$FLOW = ***** GX | \$SRQBJ= ***** GX |
| WK.SND= 000002 | W.KAST 000014 | W.STAT 000002     | \$LTM = ***** GX  | \$UCB = ***** GX  |
| WS.DIP= 000010 | W.LLT 000004  | W.TMP 000010      | \$MAIBX= ***** GX | \$WBLK = ***** GX |
| WS.INT= 000002 | W.LUN 000003  | W.WBL 000026      | \$REQID= ***** GX | \$SSHFT= 000001   |
| WS.KAS= 000004 | W.MBOX 000012 | X\$DBT= 000000    | \$RQNAM= ***** GX | \$\$\$\$ = 000062 |
| WS.STA= 000001 | W.RCVQ 000024 | \$CNBLK= ***** GX | \$SRDFM= ***** GX | .\$\$\$\$= 000034 |
| W.CINT 000022  |               |                   |                   |                   |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)

001036 001 (RW,I,LCL,REL,CON)

\$HIGH 000004 002 (RW,I,LCL,REL,CON)

Errors detected: 0

## \*\*\* Assembler statistics

Work file reads: 103

Work file writes: 104

Size of work file: 26043 Words ( 102 Pages)

Size of core pool: 17608 Words ( 67 Pages)

Operating system: RSX-11M/PLUS

Elapsed time: 00:00:27.51

SY:SESCON11S.V2,[131,134]SESCON11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCs/PA:1,[131,10]JV2,SESCON

```

283 000262 CALLR IOSUC ; Complete the request
284
285 ;+
286 ; I/O rundown disconnect substate table
287 ; -
288
289 000266 000 000 CLSTA: .BYTE USDON,NSDON ; CI sent
290 000270 000 002 .BYTE USDON,NSSDI ; Received CI, sent CC
291 000272 377 377 .BYTE -1,-1 ; Received CI
292 000274 000 002 .BYTE USDON,NSSDI ; Normal data transfer
293 000276 000 377 .BYTE US$DON,-1 ; Disconnect in progress
294 000300 004 002 .BYTE USDSC,NSSDI ; Disconnect pending

```

```

757 ; performed is OK)
758 .IFT ; NDF R$$PRO
759 TSTB C,FLG2(R4) ; Did verification succeed?
760 001160 105764 000032 BLE 20$; If LE, no ... reject the connection
761 001164 003412 ; (either unable to perform verification,
762 ; or verif. failed)
763 .ENDC ; NDF R$$PRO
764
765
766 001166 012765 000004 000022 10$: MOV #ERR$UOB,N$ERRC(R5) ; Set up 'object not installed' reason
767 001174 016403 000016 MOV C,BUF+2(R4),R3 ; Point to requested task name
768 001200 162703 000004 SUB #4,R3 ; ...
769 001204 CALL QUETSK ; Queue request to task
770 001210 103004 BCC 30$; If CC, successful
771
772 001212 016501 000022 20$: MOV N$ERRC(R5),R1 ; Get reason for disconnect
773 001216 CALL REJECT ; Reject the connection
774
775 001222 30$: RETURN
776
777 001224 BADERR: $IERRC IE.BAD&377 ; Bad temporary link address
778
779 .ENDC
780 .ENDC
781

```

```

55 .IF DF N$$NCT
56
57 .SBTTL Session control counter logic
58
59 ;+
60 **--CTRSES-Session control counter logic
61 This routine is called to update one of the ECL counters.
62 ;+
63 Calling sequence:
64 JSR R2,CTRSES
65 .BYTE <Counter offset>
66 .BYTE <Counter type>
67
68 ;+
69 Inputs:
70 R3 = Virtual address of LLT
71 R5 = Address of database descriptor
72
73 CTRSES::SAVRG <R0,R1> ; Save some registers
74 SAVMAP ; Save current mapping
75 MAPLLT ; Map to current llt
76 MOVBL (R2)+,R0 ; Get counter offset
77 BMI 10$; If MI, not a logical link counter
78 MOV L,CTR(R3),R1 ; Get address of counter block
79 BEQ 20$; If EQ, none allocated ... don't count
80 ADD R1,R0 ; Compute offset in counter block
81
82 MAP N$ENC+4(R5) ; Map to the counter blocks
83
84 10$: MOVBL (R2),R1 ; Get the counter type code
85 BIC #^C<6>,R1 ; Isolate code
86 CALL @CTRBL(R1) ; Dispatch to counting routine
87
88 20$: RESMAP ; Restore mapping
89
90 TSTB (R2)+ ; Skip over counter code
91 RESRG <R1,R0> ; Restore registers
92 RTS R2
93
94 ;+
95 ; Session control counter dispatch table
96 ;+
97 CTRTBL: .WORD ENC1W ; ECL node counter - 1 word
98 .WORD ENC2W ; ECL node counter - 2 words
99 .WORD ENCSR ; ECL node counter - bytes sent/received
100 .WORD FNCCIR ; ECL node counter - ignored CI's

```



8.13  
SESDAT - Session control local MACRO V05.03b Friday 28-Jun-85 19:54  
Table of contents

|    |     |                        |
|----|-----|------------------------|
| 6- | 42  | Macro definitions      |
| 7- | 48  | Local data             |
| 8- | 108 | Local buffers          |
| 9- | 128 | Executive vector table |

SM

SM

SM

SESDMO - Session control dissmou MACRO V05.03b Friday 28-Jun-85 19:55 Page 6  
Macro definitions

```
42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,MAPLLT,EVT$,CALLE
45 .MCALL ECDDDB$,EVLDF$,LLTDF$,MSGDF$
46
47 000000 ECDDDB$; Define ECL database offsets
48 000000 EVLDF$; Define event logging symbols
49 000000 LLTDF$; Define LLT offsets
50 000000 MSGDF$; Define message symbols
51
52 000C01 N$$SES = 1 ; This module is part of session control
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

.TITLE SESDSP - Session control dispatching  
 .IDENT /V05.00/  
 .ENABL LC

: Copyright (C) 1982, 1983, 1985 by  
 : Digital Equipment Corporation, Maynard, MASS.

: This software is furnished under a license for use only on a  
 : single computer system and may be copied only with the  
 : inclusion of the above copyright notice. This software, or  
 : any other copies thereof, may not be provided or otherwise  
 : made available to any other person except for use on such  
 : system and to one who agrees to these license terms. Title  
 : to and ownership of the software shall at all times remain  
 : in DEC.

: The information in this document is subject to change without  
 : notice and should not be construed as a commitment by Digital  
 : Equipment Corporation.

: DEC assumes no responsibility for the use or reliability of  
 : its software on equipment which is not supplied by DEC.

: Module description

Session control dispatching

: Ident history:

: 4.00 07-NOV-83  
 : DECNET-11M V4.0  
 : DECNET-11M-PLUS V2.0  
 : 5.00 22-JUL-85  
 : DECnet-11M/S V4.2  
 : DECnet-11M-Plus V3.0  
 : DECnet-Micro/RX V1.0

AXDSPP - AUXILLIARY PROCESS DIS MACRO V05.03b Tuesday 03-Sep-85 11:00<sup>C 1</sup> Page 10-3  
Symbol table

\$MDCIN 000026RG .\$\$\$\$ = 000034  
. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000514 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 16133 Words ( 64 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:35.50  
DB2:AXDSPP.T47,[131,134]AXDSPP/CR/-SP=DB2:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]T47,AXDSP

```

259 .SBTTL POST ABORT COMPLETION TO DLC LEVEL
260 +
261 **-$PABO-POST ABORT COMPLETION TO DLC LEVEL
262 POST A COMPLETION ON THE WAITING CONNECT WITH AN ABORT STATUS.
263 SET NEW LINE STATE TO DISCONNECT DELAY.
264 -
265 :
266 :
267 :
268 :
269 :
270 :
271 :
272 :
273 :
274 000216 010446 $PABO:: MOV R4,-(SP) ; SAVE R4
275 000220 012703 100362 MOV #CE.ABO,R3 ; SET ABORT STATUS
276 000224 112702 000016 MOVB #ST.DDL,R2 ; SET NEW STATE TO DISCONNECT DELAY
277 000230 012701 012000 MOV #FS.ENB,R1 ; SET SUBFUNCTION CODE
278 000234 CALL $$SCHPR ; POST COMPLETION TO DLC LEVEL
279 000240 012604 MOV (SP)+,R4
280 000242 RETURN
281 ;

```

```

630 .SBTTL ST.ABO - POST ABORT COMPLETION TO THE DLC LEVEL
631 ;+
632 ;**-.PABO-POST ABORT COMPLETION TO DLC LEVEL
633 ;
634 ; A LINE WAITING FOR CONNECT WAS DISABLED AND THE MODEM CONTROLLER
635 ; WAS UNABLE TO PICK THE RESOURCES TO NOTIFY THE DLC PROCESS. ATTEMPT
636 ; TO NOTIFY THE DLC PROCESS NOW.
637 ;
638 ;
639 ; INFUTS:
640 ; R2 - POINTER TO SYSTEM LINE TABLE
641 ; R4 - SYSTEM LINE NUMBER
642 ; R5 - POINTER TO LINE ENTRY IN MDC DATABASE
643 .PABO: CALL $PABO ; POST ABORT COMPLETION TO DLC
644 BCC .DISD ; IF SUCCESSFUL PERFORM COMPLETION TASKS
645 BISB #MC.CCB, M.CSV(R5) ; ELSE MARK ALLOCATION FAILURE
646 10$: RETURN
647 ;
648 .DSABL LSB
649 ;

```

AXSCH MACRO V05.03b Friday 28-Jun-85 18:30 Page 5  
Macro definitions

```

53 .SBTTL Macro definitions
54
55 ;
56 ; MACRO LIBRARY CALLS
57 ;
58 .MCALL INHIB$,ENABL$,SAVRG,RESRG
59 .MCALL CCBDF$,PDVDF$,OPTDF$
60 000000 ; DEFINE THE CCB OFFSETS
61 000000 ; DEFINE CEX OPTIONS
62 000000 ; DEFINE THE PDV OFFSETS
63
64 .IF DF R$$MPL
65
66 .MCALL SCBDF$
67 SCBDF$; DEFINE SCB OFFSETS
68
69 .END(

```



```

95 .IF DF R$$MPL
96
97 .SBTTL $ENUMR - Load UMRs on device enable
98
99 ;+
100 ***-$ENUMR-Load UMRs on device enable
101
102 This routine is called from the 'enable line' processing routine of
103 DDM processes. It will load the umr's for the processor on which it
104 is running if they are not already loaded.
105
106 Inputs:
107 R4 - enable ccb
108
109 ;+
110
111 $ENUMR::.IF DF R$$MPL
112 .IF NDF R$$PRO
113
114 BIT #F2.MP,@FMSK2 ; Is this a multi-processor?
115 BEO 110$; BR if no
116 TST @PUMR ; are the umr's loaded?
117 BNE 100$; yes ... don't load them again
118 SAVRG <R0,R1,R2,R3,R4,R5> ; no ... save some registers
119 MOV @UMRPT,R2 ; get pointer to umr allocation block
120 MOV M.UMRA(R2),R3 ; get first umr address to load
121 SUB #UBMPR,R3 ; get as offset
122 ASH #5,R3 ; shift to high bits
123 SUB @QSTRT,R3 ; bias correctly for pool
124 MOV R3,@PUMR ; set up new $pumr value
125 MOV M.BFVL(R2),R3 ; get pool partition address
126 MOV M.BFVH(R2),R4 ; ...
127 MOV M.UMRA(R2),R5 ; get first umr address to load
128 MOV M.UMRN(R2),R2 ; and (# of umr's to load * 4)
129 ASH #-2,R2 ; get actual number of umr's
130 10$: MOV R3,(R5)+ ; load a umr
131 MOV R4,(R5)+ ; doubleword register
132 ADD #20000,R3 ; move on by 4k
133 ADC R4 ; ...
134 SOB R2,10$; loop till all loaded
135
136 CALL $SMP1S ; start processor dependant timer
137
138 RESRG <R5,R4,R3,R2,R1,R0> ; restore registers
139 100$: INC $ACTLN ; update count of active lines on this processor
140 110$: RETURN
141
142 .ENDC
143 .ENDC
144
145 ;

```

AXSUB      CREATED BY    MACRO    ON 28-JUN-85 AT 18:31      PAGE 1      C 6

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE      | REFERENCES                                                                                         |
|---------|------------|----------------------------------------------------------------------------------------------------|
| ALOCB   | = ***** GX | 8-215                                                                                              |
| CUBGT   | = ***** GX | 12-496                                                                                             |
| CEACC   | = ***** GX | 8-219      8-231      9-281      9-293      9-310      10-380      10-392                          |
| CEDIV   | = ***** GX | 11-460                                                                                             |
| CS.ABO  | = 000100   | 12-528                                                                                             |
| CS.ERR  | = 100000   | 12-528                                                                                             |
| C.BUF1  | = 000014   | *12-521                                                                                            |
| C.LIN   | = 000006   | 12-500                                                                                             |
| DEACB   | = ***** GX | 9-275                                                                                              |
| EXRQF   | = ***** GX | 12-522                                                                                             |
| FC.MLD  | = 000026   | 12-502                                                                                             |
| FMSK2   | = ***** GX | 10-372      10-384      10-397                                                                     |
| FS.RLB  | = 002000   | 12-502                                                                                             |
| F2.DAS  | = ***** GX | 10-372      10-384      10-397                                                                     |
| I\$SAS  | = *****    | 5-62                                                                                               |
| KISAR6  | = ***** GX | 8-205      *8-235      *9-276      9-283      9-300      *9-301      *9-303      9-308      *9-314 |
|         |            | *9-316      9-323      *9-324      *9-326      *9-330      *9-337      *12-511      *12-511        |
| K\$SDAS | = *****    | 4-1                                                                                                |
| MAXLEN  | = 000006   | #5-01      12-504                                                                                  |
| MLDTS   | = 000014 R | #5-90      12-492                                                                                  |
| M\$MGE  | = 000000   | 5-66      8-198      9-267      10-370      12-507                                                 |
| N\$SVCT | = *****    | 8-205      8-235      9-276      9-283      9-300      9-301      9-303      9-308      9-314      |
|         |            | 9-316      9-323      9-324      9-326      9-330      9-337      12-511                           |
| RDBNM   | = ***** GX | 11-457                                                                                             |
| RDBTH   | = ***** GX | 11-458                                                                                             |
| R\$MPL  | = *****    | 6-95                                                                                               |
| R\$11D  | = *****    | 5-62                                                                                               |
| R\$11M  | = 000000   | 5-62                                                                                               |
| R\$11S  | = *****    | 5-62                                                                                               |
| SQRCM   | = ***** GX | 11-447      11-461                                                                                 |
| SRSTD   | = ***** GX | 12-493                                                                                             |
| XAVL    | = ***** GX | 8-209      8-236      9-288      9-332                                                             |
| XAVLL   | = 000002 R | #5-69      8-206      8-236      9-284      9-332                                                  |
| X\$MCB  | = *****    | 5-62                                                                                               |
| ZF.COU  | = 001000   | #5-62                                                                                              |
| ZF.DDM  | = 000001   | #5-62                                                                                              |
| ZF.DIA  | = 004000   | #5-62                                                                                              |
| ZF.DLC  | = 000002   | #5-62                                                                                              |
| ZF.DVP  | = 100000   | #5-62                                                                                              |
| ZF.INI  | = 0+0000   | #5-62                                                                                              |
| ZF.KMX  | = 000020   | #5-62                                                                                              |
| ZF.LLC  | = 000004   | #5-62                                                                                              |
| ZF.LMC  | = 000100   | #5-62                                                                                              |
| ZF.MAN  | = 020000   | #5-62                                                                                              |
| ZF.MFL  | = 000010   | #5-62                                                                                              |
| ZF.MTM  | = 000400   | #5-62                                                                                              |
| ZF.MUX  | = 000040   | #5-62                                                                                              |
| ZF.PSE  | = 002000   | #5-62                                                                                              |
| ZF.SLI  | = 010000   | #5-62                                                                                              |
| ZF.TIM  | = 000200   | #5-62                                                                                              |
| ZF.Y3P  | = 000000   | #5-62                                                                                              |
| ZS.ASN  | = 100000   | #5-62                                                                                              |

```

303 .IF DF R$$MPL
304 .IF NDF R$$PRO
305
306 ;+
307 ;**--MP100M-100 MSEC TIMER SERVICE (MULTI-PROCESSOR ENTRY)
308 ;
309 ; THIS ROUTINE IS CALLED EVERY 100 MSEC (WHEN ACTIVE) ON THE PROCESSOR
310 ; TO WHICH THE DEVICE REQUESTING SERVICE IS CONNECTED. IT WILL
311 ; DISPATCH TIMEOUTS TO THOSE PROCESSES REQUESTING SHORT TIMER SERVICE
312 ; WITH A SUBFUNCTION CODE OF 'FS.STM'.
313 ; -
314
315 MP100M::MOV M100Q,R0 ; POINT TO SHORT TIMER LISTHEAD
316 CALL SCANST ; SCAN SHORT TIMER QUEUE
317
318 TST @M100Q ; ANY SHORT TIMERS ACTIVE?
319 BEQ 10$; IF EQ, NO
320 MOVB @PROC2,R0 ; GET PROCESSOR NUMBER *2
321 ADD STMTB,R0 ; POINT TO ADDRESS OF TIMER QUEUE ELEMENT
322 MOV (R0),R0 ; GET ADDRESS OF TIMER QUEUE ELEMENT
323 MOV #MP100M,C.SUB(R0); SET UP ADDRESS OF PROCESSING ROUTINE
324 MOV @TK100,R2 ; GET # OF CLOCK TICKS IN 100 MSEC
325 CALLR $SMPTM ; START THE SHORT TIMER RUNNING
326
327 .ENDC
328
329 .DSABL LSB

```

.TITLE SESCON - Session control connect processing  
.IDENT /V05.00/  
.ENABL LC

Copyright (C) 1982, 1983, 1985 by  
Digital Equipment Corporation, Maynard, MASS.

This software is furnished under a license for use only on a single computer system and may be copied only with the inclusion of the above copyright notice. This software, or any other copies thereof, may not be provided or otherwise made available to any other person except for use on such system and to one who agrees to these license terms. Title to and ownership of the software shall at all times remain in DEC.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

DEC assumes no responsibility for the use or reliability of its software on equipment which is not supplied by DEC.

#### Module description

Session control connect processing

#### Ident history:

4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0

5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                            |
|---------|------------|---------------------------------------|
| ACC     | 000300 R   | 7-109 #9-186                          |
| ADDGNQ  | = ***** GX | 11-415                                |
| ADDMAI  | = ***** GX | 11-412                                |
| CATS    | = ***** GX | 10-249 10-252                         |
| CM.CON  | = 000200   | 10-268 10-301 10-315 10-319 11-394    |
| CON     | 000106 R   | 7-108 #8-130                          |
| CONTRL  | 000000 R   | 7-91 #7-108                           |
| CPYOPT  | = ***** GX | 8-156 9-189                           |
| CX.RUI  | = 000040   | 10-280                                |
| C.BUF   | 000014     | 10-240 10-255                         |
| C.BUF2  | 000024     | 10-320                                |
| C.CNT2  | 000030     | *10-299                               |
| C.FLG2  | 000032     | 10-228 *10-300 10-321                 |
| C.FNC   | 000010     | *10-268 *10-301 10-315 *10-319 11-394 |
| C.MOD   | 000011     | *10-280                               |
| C.NSP   | 000004     | *11-411                               |
| CSTA    | = ***** GX | 8-152                                 |
| DECT    | = ***** GX | 10-227                                |
| D\$LNUM | = 000014   | 10-228                                |
| ER\$ACC | = 000042   | 10-325                                |
| ER\$MLB | = 000006   | 11-390                                |
| ER\$NSR | = 000003   | 10-224                                |
| ER\$UOB | = 000004   | 10-232                                |
| IE.BAD  | = ***** GX | 9-202                                 |
| IE.NRJ  | = ***** GX | 8-165                                 |
| IE.RSU  | = ***** GX | 9-200                                 |
| IODUN   | = ***** GX | 8-167                                 |
| IOERR   | = ***** GX | 9-200 9-202                           |
| IOSUC   | = ***** GX | 9-196                                 |
| I.FCN   | = ***** GX | 7-88                                  |
| I.PRM   | = ***** GX | 8-132 8-133 8-157 9-190               |
| I.TCB   | = ***** GX | 8-149                                 |
| KISAR6  | = ***** GX | *7-95 *8-132 *8-144                   |
| LF.MMF  | = 000200   | 7-87                                  |
| L.SEGZ  | 000076     | 9-195                                 |
| L.WIND  | 000040     | *7-97                                 |
| MAPOBJ  | = ***** GX | 10-257                                |
| M.MAX   | 000011     | 11-399 11-401                         |
| M.NEXT  | 000002     | 11-384 11-385                         |
| M.RESP  | 000016     | 7-77                                  |
| M.TASK  | 000004     | 11-387                                |
| M.USE   | 000010     | *8-162 11-401 *11-409                 |
| NC.FM1  | = 000001   | 8-145                                 |
| NF\$ACC | = 100000   | 10-288                                |
| NF\$DMO | = 000010   | 10-225                                |
| NF\$RST | = 000002   | 10-230                                |
| NF.ACC  | = 000001   | 10-286                                |
| NT.CON  | = 000001   | 10-268 10-315 10-319                  |
| NT.VFY  | = 000007   | 10-301 11-394                         |
| N\$ERRC | 000022     | 8-166 *10-224 *10-232 *10-25 11-390   |
| N\$FLG  | C 00005    | 10-225                                |
| N\$LLT  | 000026     | 7-96                                  |

```

296 .SBTTL Specify network data AST
297
298
299
300
301
302
303
304
305
306
307
308
309
310 000172 .PSECT $HIGH
311
312 000172 016364 000000G 000012 SPA: MOV I.PRM(R3),M.SPA(R4)
313 000200 062704 000014 ADD #M.MAIL,R4 ; Point to mail listhead
314 000204 005001 CLR R1 ; Initialise count
315
316 000206 011404 10$: MOV (R4),R4 ; Get next queue item
317 000210 001411 BEQ 30$; If EQ, end of list
318 000212 105764 000010 TSTB C.FNC(R4) ; Is this a connect-type CCB?
319 000216 100004 BPL 20$; If PL, no ... always count it
320
321 000220 132764 000004 000011 BITB #CX.UNL,C.MOD(R4) ; Is this request pending?
322 000226 001367 BNE 10$; If NE, yes ... don't count it
323
324 000230 005201 20$: INC R1 ; Increment count of items on the queue
325 000232 000765 BR 10$; and loop
326
327 000234 012700 000000C 30$: MOV #IS.SUC&377,R0 ; Return success
328 000240 CALLR IODUN ; Complete the request

```

```

783 .SBTTL Get local node parameters
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799 001232
800 001240 016301 000002G
801 001244 017702 000000G
802 001250 062702 000006
803 001254 012704 000006
804 001260 020463 000004G
805 001264 101402
806 001266 016301 000004G
807
808 001272 160463 000004G 10$:
809 001276 112221 20$:
810 001309
811
812 001304 012700 000000C
813 001310 022763 000002 000004G
814 001316 101016
815
816 001320 017704 000000G
817 001324 116421 000036
818 001330 116421 000037
819
820 001334 022763 000004 000004G
821 001342 101004
822 001344 016421 000014
823
824 001350 012700 000000L
825 001354 166301 000002G 30$:
826 001360
827
828 000001 .END

```

+  
 \*\*--GLN--Get local node parameters  
 Fill the user's buffer with the local node name and the default segment size.  
 Inputs:  
 R1 = Address of the task's header  
 R2 = Subfunction code/4  
 R3 = Address of I/O packet  
 R4 = Address of mailbox  
 R5 = Address of database descriptor  
 The task header is mapped (RSX-11M-Plus ONLY)  
 GLN: MAP I.PRM(R3) ; Map to the user's buffer  
 MOV I.PRM+2(R3),R1 ; Get virtual address of user's buffer  
 MOV @DECPT,R2 ; Point to the local node name  
 ADD #D\$LNAM,R2 ; ...  
 MOV #6,R4 ; Assume 6 bytes to copy  
 CMP R4,I.PRM+4(R3) ; Enough space in user's buffer?  
 BLOS 10\$ ; If LOS, yes  
 MOV I.PRM+4(R3),R4 ; Use user specified size  
 10\$: SUB R4,I.PRM+4(R3) ; # of bytes remaining  
 20\$: MOVB (R2)+,(R1)+ ; Copy node name to user buffer  
 SOB R4,20\$ ; ...  
 MOV #1S.DA0&377,R0 ; Assume data overrun  
 CMP #2,I.PRM+4(R3) ; Enough room for segment size?  
 BHI 30\$ ; If HI, no  
 MOV @DECPT,R4 ; Get the DEC home block  
 MOVB D\$SEG(R4),(R1)+ ; Plant default segment size in buffer  
 MOVB D\$SEG+1(R4),(R1)+ ; ...  
 CMP #4,I.PRM+4(R3) ; Enough room for segment and node number ?  
 BHI 30\$ ; If HI, no  
 MOV D\$LNAM(R4),(R1)+ ; Else, fill in the node number  
 30\$: MOV #1S.SUC&377,R0 ; Set success code  
 SUB I.PRM+2(R3),R1 ; Compute # of bytes transferred  
 CALLR IODUN ; Complete the request  
 .END

```

102 .SBTTL ECL node counter routines
103
104 ;+
105 ;*-EMCXX-ECL node counter routines
106
107 ; These routines are entered to update ECL node counters.
108
109 ; Inputs:
110 ; R0 = Address of word in counter block
111 ; R5 = Address of database descriptor
112
113 ; Registers modified:
114 ; R0
115
116 .ENABL LSB
117
118 ;+
119 ; ECL node counter - 1 word
120 ;-
121
122 000076 005210 ENC1W: INC (R0) ; Increment counter
123 000100 001001 BNE 10$; If NE, no overflow
124 000102 005310 DEC (R0) ; Don't wrap around
125 000104
126
127 10$: RETURN
128
129 ;+
130 ; ECL node counter - 2 words
131 ;-
132
133 000106 062720 000001 ENC2W: ADD #1,(R0)+ ; Increment first word
134 000112 006402 BR 20$; Enter common code
135
136 ;+
137 ; ECL node counter - bytes sent/received
138 ;-
139
140 000114 066720 000000G ENCSR: ADD $BYTE,(R0)+ ; Add byte count to first word
141 000120 005510 20$: ADC (R0) ; Add carry into second word
142 000122 103003 BCC 30$; If CC, no overflow
143 000124 012710 MOV #1,(R0) ; Latch counter at max value
144 000130 011040 MOV (R0),-(R0) ; ...
145 000132
146
147 30$: RETURN
148
149 ;+
150 ; ECL node counter - ignored CI's
151 ;-
152
153 000134 005265 000034 ENCCIR: INC N$CIR(R5) ; Update count of ignored CI's
154 000140 001002 BNE 40$; If NE, no overflow
155 000142 005365 DEC N$CIR(R5) ; Latch counter on overflow
156 000146
157
158 40$: RETURN
159
160 .DSABL LSB
161
162 .ENDC

```



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

.TITLE SES DAT - Session control local database  
.IDENT /V05.00/  
.ENABL LC

Copyright (C) 1982, 1983, 1985 by  
Digital Equipment Corporation, Maynard, MASS.

This software is furnished under a license for use only on a  
single computer system and may be copied only with the  
inclusion of the above copyright notice. This software, or  
any other copies thereof, may not be provided or otherwise  
made available to any other person except for use on such  
system and to one who agrees to these license terms. Title  
to and ownership of the software shall at all times remain  
in DEC.

The information in this document is subject to change without  
notice and should not be construed as a commitment by Digital  
Equipment Corporation.

DEC assumes no responsibility for the use or reliability of  
its software on equipment which is not supplied by DEC.

Module description

Session control local database

Ident history:

4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0  
5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RX V1.0

SESDIS - Session control discon MACRO V05.03b Friday 28-Jun-85 19:55<sup>C 14</sup>  
Table of contents

|    |     |                                              |
|----|-----|----------------------------------------------|
| 6- | 54  | Disconnect and connect reject Q10 processing |
| 7- | 83  | Disconnect/abort Q10 processing              |
| 8- | 125 | Connect reject Q10 processing                |

```

54 .SBTTL Dismount QIO processing
55 ;+
56 ;*- $DMQIO-Dismount QIO processing
57 ;
58 ; This routine processes network dismount requests.
59 ;
60 ; Inputs:
61 ; R1 = Address of task's header
62 ; R2 = I/O subfunction code/4
63 ; R3 = Address of I/O packet
64 ; R5 = Address of database descriptor
65 ;
66 .PSECT $HIGH
67
68 000000 152765 000010 000005 $DMQIO: BLSB #NF$DMO,N$FLG(R5) ; Set 'dismount in progress' flag
69 000006 142765 000004 000005 BICB #NFSHU,NFLG(R5) ; Clear 'shut' state flag
70 000014 132763 000010 000000G BITB #10,I.FCN(R3) ; Should the state go directly to off?
71 000022 001416 BEQ 10$; If EQ, yes
72 000024 016700 000000G MOV $UCB,R0 ; Set the 'dismount in progress' flag
73 000030 152760 000000G 000000G BISB #US.MDM,U.STS(R0) ;
74 000036 EVT$ 2,0,,REOPR,SCSHU*400!S($ON) ; Issue state change event
75 000052 152765 000004 000005 BISB #NFSHU,NFLG(R5) ; Set 'shut' state flag
76
77 000060 10$: .IF DF N$SBUF
78 MOV I.TCB(R3),R0 ; Get the TCB address
79 CMP T.NAM(R0),#*RNMV ; Do buffered I/O for NMVACP only
80 BEQ 15$; If EQ, do buffered completion
81 CALL IOSUC ; Else, do regular I/O completion
82 BR 20$; and continue
83 15$: CLR I.PRM+12(R3) ; Set up $QUEBF for I/O status mapped
84 MOV I.PCB(R0),I.PRM+16(R3) ; and NMVACP's TCB address
85 MOV #IS.SUC&3?,R0 ; Return success
86 CLR R1 ;
87 SAVRG <R5> ; Save the DDB address
88 CALL @QUEBF ; Complete the I/O
89 RESRG <R5> ; Recover the DDB address
90 .IFF
91 000060 CALL IOSUC ; Complete the I/O request
92 .ENDC
93
94 000064 016504 000052 20$: MOV N$GENQ(R5),R4 ; Get next entry on general delivery queue
95 000070 001407 BEQ 30$; If EQ, no more
96 000072 011465 000052 MOV (R4),N$GENQ(R5) ; Unlink CCB from queue
97
98 000076 012701 000003 MOV #ER$NSR,R1 ; Reason = node shutting down
99 000102 CALL REJECT ; Reject the connection
100 000106 000766 BR 20$; Try next entry
101
102 000110 132765 000004 000005 30$: BITB #NF$SHU,N$FLG(R5)
103 000116 001002 BNE 100$; If NE, allow links to be disconnected
104 000120 CALLR FL$LNK ; Flush any active logical links
105
106 000124 100$: RETURN

```

SESDSP - Session control dispat MACRO V05.03b Friday 28-Jun-85 19:56 Page 6  
Macro definitions

```

42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,MAPLLT,RECMAP
45 .MCALL CCBDF$,ECDDB$,MBXDF$,LLTDF$,CNBDF$,DHBDF$
46
47 000000 CCBDF$; Define CCB offsets
48 000000 DHBDF$; Define DEC home block offsets
49 000000 ECDDB$; Define ECL database offsets
50 000000 MBXDF$; Define mailbox offsets
51 000000 LLTDF$; Define LLT offsets
52 000000 CNBDF$; Define connect pending block offsets
53
54 000001 N$$SES = 1 ; This module is part of session control

```

AXDSPP      CREATED BY    MACRO    ON 3-SEP-85 AT 11:00      PAGE 1      D 1  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE      | REFERENCES            |
|--------|------------|-----------------------|
| ASCOMP | = ***** GX | 8-168                 |
| CB.CCB | = 000002   | 10-306                |
| CCBRET | 000030 R   | 7-106 #7-120          |
| CCBRT  | = ***** GX | 7-120 10-308          |
| CE.DIS | = 100366   | 8-167                 |
| CS.LST | = 040000   | 10-303                |
| C.ADD  | 000034     | 10-270                |
| C.BID  | 000003     | 10-306                |
| C.CNT2 | 000030     | 10-274 10-286         |
| C.STS  | 000012     | *10-303               |
| DDFNC  | = ***** GX | 9-228                 |
| DUMMY  | 000026 R   | #7-119                |
| EXRQN  | = ***** GX | 10-304                |
| IS\$AS | = *****    | 5-1 7-115 9-235 9-248 |
| KMCL   | 000300 R   | 9-237 #9-249          |
| LF.ACT | = 100000   | #6-77 8-154           |
| LF.BRO | = 000400   | #6-77 10-277          |
| LF.BWT | = 000007   | #6-77                 |
| LF.ENA | = 002000   | #6-77                 |
| LF.LPB | = 001000   | #6-77                 |
| LF.MDC | = 000100   | #6-77                 |
| LF.MFL | = 004000   | #6-77                 |
| LF.MTP | = 000020   | #6-77 10-279          |
| LF.PAC | = 000200   | #6-77                 |
| LF.RDY | = 040000   | #6-77                 |
| LF.REA | = 010000   | #6-77                 |
| LF.SER | = 000040   | #6-77 10-281          |
| LF.TIM | = 000010   | #6-77                 |
| LF.UNL | = 020000   | #6-77                 |
| LF.X2P | = 000000   | #6-77                 |
| LN.CLO | = 000000   | #6-77                 |
| LN.DUM | = 000005   | #6-77                 |
| LN.LOA | = 000004   | #6-77                 |
| LN.LOO | = 000003   | #6-77                 |
| LN.OAU | = 000003   | #6-77                 |
| LN.OFF | = 000001   | #6-77                 |
| LN.ON  | = 000000   | #6-77                 |
| LN.OOP | = 000004   | #6-77                 |
| LN.OPE | = 000001   | #6-77                 |
| LN.REF | = 000002   | #6-77                 |
| LN.SER | = 000002   | #6-77                 |
| LN.STA | = 000017   | #6-77                 |
| LN.SUB | = 000360   | #6-77                 |
| LN.TRI | = 000006   | #6-77                 |
| L.COST | 000015     | #6-77                 |
| L.CTL  | 000012     | #6-77                 |
| L.CVA  | 177776     | #6-77                 |
| L.DDM  | 000002     | #6-77 9-227           |
| L.DDS  | 000004     | #6-77 9-226           |
| L.DLC  | 000003     | #6-77                 |
| L.DLM  | 000006     | #6-77                 |
| L.DLS  | 000010     | #6-77                 |

```

283 .IF NDF R$$11D & I$$AS
284 .SBTTL MODEM CONTROLLER INTERRUPT ROUTINE
285 ;+
286 **-$MDCIN-MODEM CONTROLLER INTERRUPT ROUTINE
287 :
288 THIS ROUTINE IS USED BY THOSE DRIVERS WHICH ENABLE MODEM INTERRUPTS.
289 IT IS USED TO NOTIFY THE MODEM CONTROLLER OF A CHANGE IN THE MODEM
290 STATUS.
291 :-
292 INPUTS:
293 R3 - CURRENT STATUS SERVICE BITS
294 R4 - LOW BYTE CONTAINS SYSTEM LINE #
295 :
296 000244 010546 $MDCIN::MOV R5,-(SP) ; GET A FREE REGISTER
297 000246 017705 MOV @PDVTA,R5 ; GET POINTER TO AUX PDV
298 000252 011505 MOV (R5),R5 ; ...
299 000254 016505 MOV Z.DAT(R5),R5 ; THEN POINT TO DATABASE
300 000260 105715 10$: TSTB (R5) ; HAVE WE SCANNED ALL OF THE TABLE?
301 000262 100410 BMI 30$; YES ... IGNORE THIS CALL
302 000264 126504 CMPB M.LIN(R5),R4 ; IS THIS THE CORRECT ENTRY?
303 000270 001403 BEQ 20$; YES
304 000272 062705 ADD #M.LEN,R5 ; NO ... POINT TO NEXT ENTRY
305 000276 000770 BR 10$; AND TRY AGAIN
306 000300 110365 20$: MOVB R3,M.CSV(R5) ; SET UP CURRENT SERVICE BITS
307 000304 012605 30$: MOV (SP)+,R5 ; RESTORE THE REGISTER
308 000306
309
310 .ENDC
311 ;

```

```

651 .SBTTL ST.DDL - DISCONNECT DELAY
652 +
653 : **-.DISD-DISCONNECT DELAY
654 :
655 : DELAY DISCONNECT TO ALLOW CIRCUITS TO RESET.
656 :
657 : INPUTS:
658 : R2 - POINTER TO SYSTEM LINE TABLE
659 : R4 - SYSTEM LINE NUMBER
660 : R5 - POINTER TO LINE ENTRY IN MDC DATABASE
661 :
662 001040 105765 000002 .DISD: TSTB M.TIM(R5) ; IS THE TIMER ACTIVE?
663 001044 001014 BNE 10$; YES ... KEEP WAITING
664 001046 012703 000040 MOV #CS.DIS,R3 ; SET COMPLETION STATUS
665 001052 112702 000000 MOVB #ST.IDL,R2 ; SET NEW STATUS
666 001056 012701 013000 MOV #FS.DIS,R1 ; SET SUBFUNCTION CODE
667 001062 CALL $SCHPR ; AND SCHEDULE DLC PROCESS
668 001066 103003 BCC 10$; ALLOCATION FAILURE?
669 001070 152765 000200 000004 BISB #MC.CCB, M.CSV(R5) ; YES...MARK ALLOCATION FAILURE
670 001076 10$: RETURN
671 ;

```

71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127

.SBTIL \$\$QSRV - GENERALIZED SOFTWARE LEVEL SERVICE

\*\*\*-\$\$QSRV-GENERALIZED SOFTWARE LEVEL SERVICE

INPUTS:

R3 = ADDRESS OF FORK BLOCK + 2

OUTPUTS:

A CCB CHAIN IS DEQUEUED FROM THE LEVEL QUEUE AND DISPATCHED TO THE APPROPRIATE PROCESS. ALL ENTRIES ON THE LEVEL QUEUE ARE REMOVED BEFORE THE LEVEL IS EXITED.

NOTE:

DISPATCH IS MADE TO A DLC OR AN LLC PROCESS BASED ON THE SENSE OF THE MSB OF THE WORD DEFINED BY C.LIN. IF THE MSB IS ZERO, THE CELL CONTAINS A SLN & STATION NUMBER PAIR AND A DLC IS DISPATCHED USING THE SLN. IF THE MSB IS ONE, THE CELL CONTAINS A PDV INDEX AND CHANNEL NUMBER PAIR AND AN LLC PROCESS IS DISPATCHED.

```

5$: MOV CMFRK,R3 ; SET ADDRESS OF FORK BLOCK + 4
 ADD #4,R3 ;...
 MTPS #PR7 ;::: DISABLE INTERRUPTS

 .IF DF R$$MPL
 .IF NDF R$$PRO

 BIT #F2.MP,@FMSK2 ;::: IS THIS A MULTIPROCESSOR?
 BEQ 6$;::: BR IF NO
 CALL @MPLCK ;::: LOCK ACCESS TO PROCESS QUEUE

6$: .ENDC
 .ENDC

 CMP @TKPS,@INTCT ;::: TOO MANY CLOCK TICKS PENDING?
 BLT 47$;::: IF LT, YES ... GIVE UP PROCESSOR

 MOV (R3),R4 ;::: GET FIRST ENTRY ON 0 BUE
 BEQ 50$;::: IF EQ NO ENTRIES, EX LEVEL

 .IF DF R$$MPL
 .IF NDF R$$PRO

 BIT #F2.MP,@FMSK2 ;::: IS THIS A MULTIPROCESSOR?
 BEQ 7$;::: BR IF NO
 MOV C.URM(R4),R2 ;::: GET REQUIRED UNIBUS RUN MASK
 BEQ 7$;::: IF ZERO, WE CAN ALWAYS EXECUTE THIS ONE
 MOV @CPURM,-(SP) ;::: GET POINTER TO CPU URM TABLE
 BIT R2,@(SP)+ ;::: CAN WE EXECUTE ON THIS PROCESSOR?
 BNE 7$;::: YES ... GO AHEAD

```



```

144 .SBTTL $DSUMR - Check for last line disabled on this processor
145 ;+
146 ;**-$DSUMR-Check for last line disabled on this processor
147 ;
148 ; If this is the last line disabled on this processor then we can mark
149 ; the UMRsas not being loaded. This will allow us to correctly handle
150 ; other processors being taken off-line.
151 ;
152 ; Inputs:
153 ; R4 - disable ccb
154 ; -
155 $DSUMR:: .IF DF R$$MPL
156 .IF NDF R$$PRO
157
158 BIT #F2.MP,@FMSK2 ; Is this a multi-processor?
159 BEQ 10$; BR if no
160 DEC $ACTLN ; reduce count of active lines on this processor
161 BNE 10$; are there any still active?
162 CLR @PUMR ; no ... umr's no longer in use
163
164 .ENDC
165 .ENDC
166
167 10$: RETURN
168
169 .ENDC
170

```

AXSUB      CREATED BY    MACRO    ON 28-JUN-85 AT 18:31      PAGE 2      D 6

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE     | REFERENCES        |
|---------|-----------|-------------------|
| ZS.BSY  | = 140000  | #5-62             |
| Z.AVL   | 000014    | #5-62             |
| Z.DAT   | 000016    | #5-62             |
| Z.DSP   | 000000    | #5-62      5-62   |
| Z.FLG   | 000010    | #5-62             |
| Z.LEN   | = 000016  | #5-62             |
| Z.LLN   | 000006    | #5-62             |
| Z.MAP   | 000020    | #5-62             |
| Z.NAM   | 000004    | #5-62             |
| Z.PCB   | 000012    | #5-62             |
| Z.SCH   | 000007    | #5-62             |
| \$ALOCX | 000020 RG | #8-197            |
| \$CMEXI | 000602 RG | #11-451           |
| \$CMINI | 000570 RG | #11-447           |
| \$DEACX | 000150 RG | #9-266            |
| \$RQSTL | 000650 RG | #12-488           |
| \$SQRTB | 000006 R  | #5-78      11-456 |
| \$XLINK | 000432 RG | #10-366           |

```

331 .SBTTL SCAN SHORT TIMER QUEUE
332
333 ***-SCANST-SCAN SHORT TIMER QUEUE
334
335 SCAN A SHORT TIMER QUEUE FOR ENTRIES WHICH HAVE EXPIRED AND DISPATCH
336 TO THE SPECIFIED PROCESSES.
337
338 INPUTS:
339 R0 - POINTER TO SHORT TIMER QUEUE LISTHEAD
340
341 REGISTERS MODIFIED:
342 R1, R2, R3, R4, R5
343
344 000300 005777 000000C SCANST: RST @PWRF1 ; IS POWERFAIL RECOVERY UNDERWAY?
345 000304 100473 ; IF MI, YES ... DON'T DISPATCH ANY TIMERS
346
347 000306 MTPS #PR7 ;;; DISABLE INTERRUPTS
348
349 .IF DF R$$MPL
350 .IF NDF R$$PRO
351
352 BIT #F2.MP,@FMSK2 ;;; IS THIS A MULTI-PROCESSOR?
353 BEQ 4$;;; BR IF NO
354 CALL @MPLCK ;;; LOCK ACCESS TO COMMEEXEC RESOURCES
355 4$:
356 .ENDC
357 .ENDC
358
359 000314 011004 MOV (R0),R4 ;;; GET LIST OF TIMER CELLS
360 000316 005010 CLR (R0) ;;; AND RESET LISTHEAD
361 000320 010060 000002 MOV R0,2(R0) ;;; ...
362
363 .IF DF R$$MPL
364 .IF NDF R$$PRO
365
366 BIT #F2.MP,@FMSK2 ;;; IS THIS A MULTI-PROCESSOR?
367 BEQ 6$;;; BR IF NO
368 CALL @ (SP)+ ;;; CO-ROUTINE RETURN TO UNLOCK RESOURCES
369 6$:
370 .ENDC
371 .ENDC
372
373 000324 10$: MTPS #0 ; RESET PRIORITY
374
375 000330 011446 MOV (R4),-(SP) ; SAVE LINK TO NEXT CELL
376 000332 SAVRG <R0> ; SAVE LISTHEAD ADDRESS
377
378 000334 005364 000004 DEC X.TMR(R4) ; REDUCE TIMER COUNT
379 000340 001046 BNE 60$; IF NE, NOT YET EXPIRED
380 000342 016464 000006 000004 MOV X.RTMR(R4),X.TMR(R4) ; RESET TIMER VALUE
381
382 000350 116401 000003 MOVB X.FLAG(R4),R1 ; GET TIMER FLAGS
383 000354 100402 BMI 20$; IF MI, TIMER HAS BEEN CANCELLED
384 000356 STALT ; REPLACE TIMER CELL ON LIST
385
386 000362 005002 20$: CLR R2 ; GET TIMER CELL ID
387 000364 156402 000002 BISB X.ID(R4),R2 ; ...

```

```

42 Macro definitions
43
44 .SBTTL Macro definitions
45 .MCALL SAVRG,RESRG,MAP,MAPLLT,RECMAP
46 .MCALL $IERRC,DHBD$
47 .MCALL CRBD$,LLWDF$,MBXDF$,ECDD$,LLTDF$,OBJDF$
48 .MCALL NSFDF$,CNBDF$,CCBDF$,NSSYM$
49 DHBD$; Define DEC home block offsets
50 CRBD$; Define connect request block offsets
51 LLWDF$; Define window block offsets
52 MBXDF$; Define mailbox offsets
53 ECDD$; Define ECL database offsets
54 LLTDF$; Define LLT offsets
55 OBJDF$; Define object block offsets
56 NSFDF$; Define features mask symbols
57 CNBDF$; Define connect pending block offsets
58 CCBDF$; Define CCB offsets
59 NSSYM$; Define NSP symbols
60
61 000001 NSSSES = 1 ; This module is part of session control

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE    | REFERENCES                            |
|---------|----------|---------------------------------------|
| N\$LLTM | 000024   | 7-95                                  |
| N\$MBXQ | 000050   | 11-384                                |
| N\$PLLT | 000030   | 7-98                                  |
| N\$SNOD | 000012   | 10-299                                |
| N\$SACC | = 000001 | 10-282 10-313 11-392                  |
| N\$SEVI | = 000001 | #4-2                                  |
| N\$SMCP | = *****  | 10-234 10-262 10-270                  |
| N\$SMLL | = 000001 | 7-95                                  |
| N\$SSES | = 000001 | #6-61                                 |
| N\$SSLI | = *****  | 7-81 11-363                           |
| N\$SVCT | = *****  | 7-95 8-132 8-144                      |
| N.DDE   | 000010   | 10-247                                |
| N.DFM   | 000004   | 10-243                                |
| N.DC    | 000005   | 10-241 10-256                         |
| N.RIDC  | 000032   | 8-135 8-140                           |
| N.RND   | 000000   | 8-135                                 |
| N.ROL   | = 000110 | 8-140                                 |
| OF.RLU  | = 000100 | 10-278                                |
| QUETSK  | = 000716 | 10-310 #11-353                        |
| RMVWND  | = *****  | 7-101                                 |
| R\$SMPL | = *****  | 11-353                                |
| R\$PRO  | = *****  | 10-293 10-304 10-318 11-404           |
| SNDACC  | = *****  | 9-192                                 |
| SNDCON  | = *****  | 8-159                                 |
| SRSTD   | = *****  | 11-359                                |
| TLCHK   | = *****  | 9-186                                 |
| T.NAM   | = *****  | 8-150 8-151                           |
| UISAR6  | = *****  | 8-144                                 |
| USRCI   | = 000360 | #10-223                               |
| U.CW3   | = *****  | 10-285                                |
| VFYNAM  | = *****  | 10-302                                |
| W.LLT   | 000004   | *7-98                                 |
| W.MBOX  | 000012   | 7-76                                  |
| W.SEGZ  | 000006   | *9-195                                |
| W.TMP   | 000010   | *8-130                                |
| \$CNBLK | = *****  | 8-134                                 |
| \$CNQIO | = 000000 | #7-76                                 |
| \$FLOW  | = *****  | *7-87 7-90                            |
| \$LTM   | = *****  | *7-77                                 |
| \$MAIBX | = *****  | *7-78 8-161 *11-410                   |
| \$REGID | = *****  | 8-139                                 |
| \$RQNAM | = *****  | *10-248 *10-250 *10-253 10-259 10-260 |
| \$SRDFM | = *****  | *8-145                                |
| \$SRDSC | = *****  | 8-147                                 |
| \$SROBJ | = *****  | *8-146                                |
| \$UCB   | = *****  | 10-284                                |
| \$WBLK  | = *****  | *7-79 7-92 9-194                      |

```

330 .SBTIL Get network data
331
332 +
333 ***GND-Get network data
334
335 Scan the network data queue for a data item which satisfies the
336 requirements of this request.
337
338 Inputs:
339 R1 = Address of the task's header
340 R2 = Subfunction code/4
341 R3 = Address of I/O packet
342 R4 = Address of mailbox
343 R5 = Address of database descriptor
344
345 .PSECT
346 000302
347 GND: CALL FNDMAI ; Find matching mail item
348 BCS 100$; If CS, none found
349
350 MOV R0,-(SP) ; Save type code on stack
351 SWAB (SP) ; Shift code to high byte
352 BISB #IS.SUC,(SP) ; Assume successful completion
353
354 MOV C.CNT(R1),-(SP) ; Set up length on the stack
355
356 CMPB R2,#10 ; Length and type only GND$?
357 BEQ 40$; If EQ, yes
358
359 CMP (SP),1.PRM+4(R3) ; Do we have enough space for the request?
360 BLOS 10$; If LOS, yes
361 MOV 1.PRM+4(R3),(SP) ; Update # of bytes transferred
362 MOV #IS.DAO,2(SP) ; Change return code to data overrun
363
364 10$: TSTB C.FNC(R1) ; Is this a connect type request?
365 BPL 20$; If PL, no ... remove entry from the queue
366 BISB #CX.UNL,C.MOD(R1) ; Mark entry as de-queued
367 BR 30$; Process request
368
369 20$: MOV (R1),(R4) ; Unlink item from list
370
371 30$: MAP 1.PRM(R3) ; Map the user's buffer
372 MOV 1.PRM+2(R3),R4 ; Get virtual address of user's buffer
373 ASL R0 ; Form word offset
374 CALL @TDISP-2(R0) ; Dispatch to processing routine
375
376 40$: MOV (SP)+,R1 ; Recover second I/O status word
377 MOV (SP)+,R0 ; Recover first I/O status word
378 CALLR IODUN ; Complete the request
379
380 100$: $IERRC IE.NDA8377 ; No data found
381
382 +
383 ; Type code dispatch table
384 -
385
386 TDISP: .WORD CON ; 1 - connect request
387 .WORD INT ; 2 - interrupt message

```

|                  |                 |                  |                  |                  |
|------------------|-----------------|------------------|------------------|------------------|
| ABO 000540R      | CM.XLO= 000004  | C.FLG1 000022    | D\$SEG 000036    | FS.REA= 001000   |
| ABT 000550R      | CM.Y 000446R    | C.FLG2 000032    | D\$SER 000032    | FS.RET= 000000   |
| ACCLLT= ***** GX | CP.DCF= 000040  | C.FNC 000010     | D\$SQL 000052    | FS.REZ= 003000   |
| ADDGNQ= ***** GX | CP.HDL= 000007  | C.LIN 000006     | D\$BUG= 177514   | FS.RLB= 002000   |
| ADDMAI= ***** GX | CP.PS= 177400   | C.LNK 000000     | D\$ISK= 000000   | FS.RNG= 011000   |
| A\$CHK= 000000   | CP.PSI= 000200  | C.MOD 000011     | D\$LL1= 000001   | FS.RST= 000000   |
| A\$CPS= 000000   | CP.XCF= 000100  | C.NSP 000004     | D\$SYNC= 000020  | FS.RTN= 001000   |
| A\$PRI= 000000   | CP.2FR= 000030  | C.PRO 000042     | D\$SYNM= 000000  | FS.SET= 005000   |
| A\$TRP= 000000   | CS.ABO= 000100  | C.RSV 000002     | ERSABM= 000010   | FS.SFC= 005000   |
| ADERR 001224R    | CS.BRO= 000002  | C.STA 000007     | ERSABO= 000046   | FS.SFR= 006000   |
| BLXIO= ***** GX  | CS.BUF= 000200  | C.STS 000012     | ERSABT= 000011   | FS.SFS= 004000   |
| CB.CCB= 000002   | CS.CES= 000002  | C.URM 177776     | ERSACC= 000042   | FS.SPW= 040000   |
| CB.DDM= 000040   | CS.CHN= 000010  | C.XACP 000004    | ERSCDI= 000052   | FS.STM= 000000   |
| CB.D.C= 000020   | CS.CMP= 000200  | C.XID 000035     | ERSCOM= 000047   | FS.STP= 002000   |
| CB.RDB= 000004   | CS.DCR= 000400  | C.XLEN 000044    | ERSFMT= 000005   | FS.STR= 001000   |
| CB.SDB= 000010   | CS.DEF= 000004  | C.XPLI 000040    | ERSMLB= 000006   | FS.TRM= 003000   |
| CB.SLI= 000100   | CS.DEV= 000002  | C.XPT 000034     | ERSNNF= 000012   | FS.WLB= 001000   |
| CB.XLB= 000001   | CS.DIS= 000040  | C.XSVC 000042    | ERSNOD= 000002   | FS.XKL= 002000   |
| CC.LLC= 000200   | CS.ENA= 000001  | C.XIC 000037     | ERSNSL= 000013   | FS.XOF= 010000   |
| CE.ABO= 100362   | CS.ENB= 000020  | C.X25 000036     | ERSNSR= 000003   | FS.XON= 007000   |
| CE.DAO= 100346   | CS.ERR= 100000  | DEACB= ***** GX  | ERSRES= 000001   | FS.ZER= 002000   |
| CE.DIS= 100366   | CS.FTL= 001000  | DECP= ***** GX   | ERSSTA= 000051   | F\$SLVL= 000001  |
| CE.ERR= 100370   | CS.HCR= 000001  | DISCMP= ***** GX | ERSUOB= 000004   | GLN 001232R      |
| CE.ILN= 100350   | CS.HFE= 002000  | DSC 000550R      | EVT 000576R      | GND 000302R      |
| CE.LTO= 100356   | CS.LST= 040000  | D\$AMXC 000072   | E\$XPR= 000000   | G\$STPP= 000000  |
| CE.MOP= 100372   | CS.MTL= 004000  | D\$AMXH 000074   | FC.CCP= 000020   | G\$STSS= 000000  |
| CE.NTE= 100361   | CS.RNG= 000010  | D\$ANN 000000    | FC.CTL= 000006   | G\$STTK= 000000  |
| CE.RTE= 100376   | CS.ROV= 000004  | D\$BRPR 000102   | FC.KCP= 000016   | G\$SWRD= 000000  |
| CE.SRC= 100364   | CS.RSN= 010000  | D\$BRTM 000100   | FC.KIL= 000004   | H.LUN= ***** GX  |
| CE.STP= 100352   | CS.SHU= 000001  | D\$DELF 000045   | FC.MAN= 000024   | H.NLUN= ***** GX |
| CE.TME= 100354   | CS.SID= 000002  | D\$DELD 000046   | FC.MLD= 000026   | H.NML= ***** GX  |
| CE.TMO= 100374   | CS.STR= 000004  | D\$END= 000104   | FC.PCT= 000030   | IE.ABO= ***** GX |
| CE.UNS= 100344   | CS.SUC= 000001  | D\$FNB 000034    | FC.PWR= 000022   | IE.BAD= ***** GX |
| CF.CHN= 000001   | CS.TMO= 020000  | D\$HIO 000024    | FC.RCE= 000002   | IE.INS= ***** GX |
| CF.EOM= 000004   | CF.XUR= 000004  | D\$HOST 000022   | FC.RCP= 000014   | IE.NDA= ***** GX |
| CF.HDR= 000020   | CILDSP 000004R  | D\$INAC 000044   | FC.TIM= 000010   | IE.RSU= ***** GX |
| CF.LB= 100000    | CV\$MSK= 000003 | D\$INCT 000042   | FC.XCP= 000012   | IE.SPC= ***** GX |
| CF.LIN= 000002   | CV\$31= 000001  | D\$IPL 000051    | FC.XME= 000000   | INT 000512R      |
| CF.SOM= 000010   | CV\$32= 000000  | D\$IID 000020    | FLSHIO= ***** GX | IN.DAT= 000400   |
| CF.SYN= 000040   | CV\$40= 000002  | D\$INAM 000006   | FLSHMB= ***** GX | IN.ILS= 000001   |
| CF.TRN= 000100   | CX.GDQ= 000001  | D\$LNUN 000014   | FNDMAI 000244h   | IODUN= ***** GX  |
| CLS 000000R      | CX.REM= 000020  | D\$SLT 000047    | FNDMBX= ***** GX | IODERR= ***** GX |
| CLSDON 000176R   | CX.REQ= 000002  | D\$MAXC 000064   | FS.AST= 000000   | IOSUC= ***** GX  |
| CLSTA 000266R    | CX.RUI= 000040  | D\$MAXH 000066   | FS.CIB= 002000   | IS.DAQ= ***** GX |
| CL\$MFL= 000010  | CX.SMC= 000010  | D\$MAXV 000070   | FS.CRA= 001000   | IS.SUC= ***** GX |
| CL\$SFL= 000004  | CX.UNL= 000004  | D\$MLL 000040    | FS.DIS= 013000   | IS\$RAR= 000000  |
| CL\$TYP= 000001  | C\$GORE= 000400 | D\$MNOD 000041   | FS.DVC= 001000   | IS\$RDN= 000000  |
| CL.MUT= 000001   | C\$SRSH= 177564 | D\$NA 000062     | FS.ENB= 012000   | I.LN2= ***** GX  |
| CL.MU2= 000002   | C.ADD 000034    | D\$NBEA 000056   | FS.EXI= 001000   | I.PRM= ***** GX  |
| CL.RES= 177774   | C.BID 000003    | D\$NBRA 000054   | FS.GET= 006000   | I.TCB= ***** GX  |
| CMPINT= ***** GX | C.BUF 000017    | D\$NEND= 000054  | FS.HLT= 000000   | KILLNK= ***** GX |
| CM.CIR= 000002   | C.BUF1 000014   | D\$NLN 000030    | FS.INI= 000000   | KISAR6= ***** GX |
| CM.CON= 000200   | C.BUF2 000024   | D\$NN 000060     | FS.KIL= 000000   | K\$CNT= 177546   |
| CM.FMT= 100000   | C.CNT 000020    | D\$OUTT 000043   | FS.LCL= 100000   | K\$CSR= 177546   |
| CM.HRD= 000002   | C.CNT1 000020   | D\$RETF 000050   | FS.LTM= 001000   | K\$SLDC= 000000  |
| CM.LIN= 000000   | C.CNT2 000030   | D\$RN 000002     | FS.MNT= 004000   | K\$TIPS= 000074  |
| CM.LOO= 000001   | C.FLG 000022    | D\$PTMR 000076   | FS.MSN= 014000   |                  |

```

157 .IF DF N$$EVL
158
159 .SBTTL Session control event logging
160
161 ;+
162 ;**--EVLSES-Session control event logging
163
164 This routine is called to log an event which has been generated
165 by session control.
166
167 Calling sequence:
168 JSR R5,EVLSES
169 .WORD <Class*64.+type>
170 .WORD <Event control mask>
171 [.WORD Parameter1] ; Optional parameters
172 [.WORD Parameter2]
173
174 Inputs:
175 R4 = Address of data to log
176
177 EVLSES::SAVRG <R0,R1,R2,R3,R4>
178 000150
179 000162 016703 000000G MOV EVDSC,R3 ; Point to event descriptor block
180 000166 012500 MOV (R5)+,R0 ; Get class and event type
181 000170 012501 MOV (R5)+,R1 ; Get event control mask
182 000172 010102 MOV R1,R2 ; Get copy of event mask
183 000174 100006 BPL 10$; If PL, no optional parameters
184
185 000176 012563 000002 MOV (R5)+,E$PRM(R3) ; Copy parameter
186 000202 006302 ASL R2 ; Second parameter present?
187 000204 100002 BPL 10$; If PL, no
188 000206 012563 000004 MOV (R5)+,E$PRM+2(R3)
189
190 000212 042701 140000 10$: BIC #^C<37777>,R1 ; Remove parameter count
191
192 000216 SAVRG <R5> ; Save return PC
193 000220 CALL @CELOG ; Log the event
194 000224 RESRG <R5> ; Restore calling PC
195
196 000226 RESRG <R4,R3,R2,R1,R0>
197 000240 000205 RTS R5
198
199 .ENDC
200 000001 .END

```



0.13  
SESDAT - Session control local MACRO V05.03b Friday 28-Jun-85 19:54 Page 6  
Macro definitions

```
42 .SBTTL Macro definitions
43
44 .MCALL CRBDF$
45
46 000000 CRBDF$; Define connect block offsets
```

```

1 .TITLE SESDIS - Session control disconnect processing
2 .IDENT /V05.00/
3 .ENABL LC
4
5 Copyright (C) 1982, 1983, 1985 by
6 Digital Equipment Corporation, Maynard, MASS.
7
8 This software is furnished under a license for use only on a
9 single computer system and may be copied only with the
10 inclusion of the above copyright notice. This software, or
11 any other copies thereof, may not be provided or otherwise
12 made available to any other person except for use on such
13 system and to one who agrees to these license terms. Title
14 to and ownership of the software shall at all times remain
15 in DEC.
16
17 The information in this document is subject to change without
18 notice and should not be construed as a commitment by Digital
19 Equipment Corporation.
20
21 DEC assumes no responsibility for the use or reliability of
22 its software on equipment which is not supplied by DEC.
23
24 Module description
25
26 Session control disconnect processing
27
28 Ident history:
29
30 4.00 07-NOV-83
31 DECNET-11M V4.0
32 DECNET-11M-PLUS V2.0
33
34 5.00 22-JUL-85
35 DECnet-11M/S V4.2
36 DECnet-11M-Plus V3.0
37 DECnet-Micro/RX V1.0
38
39
40
41
42
43 .MCALL SAVRG,RESRG,MAP,MAPLLT,CALLE
44 .MCALL $IERRC
45 .MCALL LLWDF$,MBXDF$,ECDDB$,LLTDF$
46
47 LLWDF$; Define window block offsets
48 MBXDF$; Define mailbox offsets
49 ECDDB$; Define ECL database offsets
50 LLTDF$; Define LLT offsets
51
52 000001 N$$SES = 1 ; This module is part of session control

```

```

108 .SBTTL Flush any active logical links
109 ;+
110 ;**-FLSLNK-Flush any active logical links
111 ;
112 ; Scan the logical link address table for any active logical links
113 ; and break them.
114 ;-
115 ; Inputs:
116 ; R5 = Address of database descriptor
117 ;
118 000000 .PSECT
119
120 000000 016501 000036 FLSLNK: MOV N$LVC(R5),R1 ; Get # of LLT's to scan
121 000004 016502 000040 MOV N$LVC+2(R5),R2 ; Get pointer to logical link table
122 000010 005722 TST (R2)+ ; Skip over first entry
123
124 000012 012203 10$: MOV (R2)+,R3 ; Get address of next LLT
125 000014 001417 BEQ 20$; If EQ, none present
126 000016 CALLE ACCLLT ; Gain access to this LLT
127
128 000026 SAVRG <R1,R2> ; Save scan state
129 000032 012763 000013, 000100 MOV #ER$NSL,L,DCR(R3)
130 000040 012702 000060, MOV #DMOTA-2,R2 ; Point to substate table
131 000044 CALL KILLNK ; Kill the logical link
132 000050 RESRG <R2,R1>
133
134 000054 20$: SOB R1,10$; Scan all logical links
135 000060 RETURN
136
137 ;+
138 ; Dismount user and network substate table
139 ;-
140
141 000062 002 000 DMOTA: .BYTE USCNF,NSDON ; CI sent
142 000064 006 000 .BYTE USDIS,NSDON ; Received CI, sent CC
143 000066 377 377 .BYTE -1,-1 ; Received CI
144 000070 006 000 .BYTE USDIS,NSDON ; Normal data transfer
145 000072 377 377 .BYTE -1,-1 ; Disconnect in progress
146 000074 004 000 .BYTE USDSC,NSDON ; Disconnect pending
147
148 000001 .END

```

```

56 .SBTTL Dispatch tables
57
58 ;+
59 ; CCB function dispatch table
60 ; -
61 000000 .PSECT $HIGH
62
63 000000 CCBDSP: .IF DF N$$$SLI
64
65 .WORD .SEXME ; Transmit enable - SLI requests
66 .WORD .SERCE ; Receive enable - SLI requests
67 .WORD .+1 ; Kill enable - illegal
68 .WORD .+1 ; Control enable - illegal
69 .WORD .+1 ; Timeout - illegal
70 .WORD .+1 ; Transmit complete - illegal
71 .WORD .SERCP ; Receive complete - session control message received
72 .WORD .+1 ; Kill complete - illegal
73 .WORD .+1 ; Control complete - illegal
74
75 .IFF
76
77 000000 000001' .WORD .+1 ; Transmit enable - illegal
78 000002 000003' .WORD .+1 ; Receive enable - illegal
79 000004 000005' .WORD .+1 ; Kill enable - illegal
80 000006 000007' .WORD .+1 ; Control enable - illegal
81 000010 000011' .WORD .+1 ; Timeout - illegal
82 000012 000013' .WORD .+1 ; Transmit complete - illegal
83 000014 000000G .WORD .SERCP ; Receive complete - session control message received
84 000016 000017' .WORD .+1 ; Kill complete - illegal
85 000020 000021' .WORD .+1 ; Control complete - illegal
86
87 .ENDC
88
89 ;+
90 ; QIO function dispatch table
91 ; -
92 000022 000000G .WORD $DMQIO ; Dismount request
93 000024 000000G .WORD $CLQIO ; Close LUN request
94 000026 000000G QIODSP: .WORD $CNQIO ; Connect request
95 000030 000000G .WORD $DSQIO ; Disconnect request
96 000032 000000G .WORD $CTQIO ; Network control request

```

AXDSPP      CREATED BY    MACRO    ON 3-SEP-85 AT 11:00      PAGE 2      E 1  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL   | VALUE      | REFERENCES                                                      |
|----------|------------|-----------------------------------------------------------------|
| L.FLG    | 000000     | #6-77                                                           |
| L.KRBA   | 000016     | #6-77                                                           |
| L.LEN    | = 000022   | #6-77      10-285                                               |
| L.MPF    | 000022     | #6-77      8-163                                                |
| L.NMST   | 000020     | #6-77                                                           |
| L.NSTA   | 000014     | #6-77      8-157      8-171                                     |
| L.OWNR   | 000021     | #6-77                                                           |
| L.UNT    | 000013     | #6-77                                                           |
| NETACP   | 000304 R   | #10-267      10-293                                             |
| NMCL2    | = ***** GX | 10-297      10-299                                              |
| NMCMF    | 000310 R   | 7-112      #10-269                                              |
| N\$1LN   | = *****    | 8-141                                                           |
| PD\$PL   | = ***** GX | 9-229                                                           |
| PW\$FAIL | 000034 R   | 7-98      #8-136      8-172      8-175                          |
| PW\$RFL  | 000170 R   | 7-108      #9-202                                               |
| PW\$RFL  | = ***** GX | 8-136      8-174      9-202      9-245                          |
| P\$RFL   | = 000001   | #4-2      5-15      7-97      7-107      8-122                  |
| RDBRT    | = ***** GX | 10-310                                                          |
| R\$MPL   | = *****    | 9-213                                                           |
| R\$11D   | = *****    | 5-1                                                             |
| SF.ACT   | = 000200   | #6-77      8-163                                                |
| SF.ENA   | = 000100   | #6-77                                                           |
| SF.LPB   | = 000004   | #6-77                                                           |
| SF.MFL   | = 000040   | #6-77                                                           |
| SF.PAC   | = 000020   | #6-77                                                           |
| SF.REA   | = 000010   | #6-77                                                           |
| SF.SER   | = 000001   | #6-77      10-290                                               |
| SF.SVC   | = 000002   | #6-77                                                           |
| SF.UNL   | = 000040   | #6-77                                                           |
| SLTMA    | = ***** GX | 8-149      9-208      10-276                                    |
| SLTNM    | = ***** GX | 9-203      9-245                                                |
| SRSTD    | = ***** GX | 9-238      10-294                                               |
| S.COST   | 000001     | #6-77                                                           |
| S.FLG    | 000000     | #6-77                                                           |
| S.LEN    | 000004     | #6-77      10-188                                               |
| S.NMST   | 000002     | #6-77                                                           |
| S.OWNR   | 000003     | #6-77                                                           |
| TRIB     | 000166 R   | 8-159      8-165      *8-170      8-171      *8-173      #8-191 |
| TSKRT    | = ***** GX | 9-241                                                           |
| X\$MDC   | = *****    | 5-8      7-88      7-115      8-177                             |
| \$AUXTB  | 000000 RG  | #7-84                                                           |
| \$BFRTN  | = ***** GX | 7-86                                                            |
| \$MDCIN  | 000026 RG  | #7-116                                                          |

```

313 .SBTTL MODEM CONTROLLER TIMER SERVICE
314
315 +
316 **MDMSCN-MODEM CONTROLLER TIMER SERVICE
317
318 THIS ROUTINE IS ENTERED ONCE PER SECOND AND SCANS THE ACTIVE LINES
319 FOR A CHANGE IN STATE OR TIMEOUT.
320
321 -
322 INPUTS:
323 R5 - POINTER TO MDC DATABASE
324
325 MDMSCN::TSTB (R5) ; HAVE WE SCANNED ALL OF THE TABLE?
326 BMB 50$; YES
327 TSTB M.STT(R5) ; IS THE LINE IDLE?
328 BEQ 40$; YES ... NOTHING FOR US TO DO
329 BITB #MS.SCA,(R5) ; DOES THIS LINE REQUIRE SCANNING?
330 BEQ 10$; NO ... MUST USE MODEM INTERRUPTS
331 MOVB M.LIN(R5),R3 ; GET THE SYSTEM LINE NUMBER
332
333 .IF DF N$S1LN
334
335 MOV @SLTMA,R2 ; GET POINTER TO SYSTEM LINE TABLE ENTRY
336 MOV (R2),R2 ; ...
337
338 .IFF
339
340 MOV R3,R2 ; COPY IT
341 ASL R2 ; FORM WORD INDEX
342 ADD @SLTMA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE
343 MOV (R2),R2 ; GET POINTER TO SYSTEM LINE TABLE ENTRY
344
345 .ENDC
346
347 TST (R2) ; IS THE LINE ACTIVE? (LF.ACT SET)
348 BPL 40$; NO ... DON'T CALL THE DDM
349
350 .IF DF RSMPL
351 .IF NDF RSPRO
352
353 BIT #F2.MP,@FMSK2 ; IS THIS A MULTIPROCESSOR SYSTEM?
354 BEQ 5$; BR IF NO
355 MOV L.KRBA(«2),R1 ; GET POINTER TO KRB
356 MOV @CPURM,«(SP) ; GET POINTER TO CPU URM TABLE
357 BIT K.URM(R1),«(SP)+ ; IS BUS ACTIVE ON THIS PROCESSOR?
358 BEQ 40$; NO ... ANOTHER PROCESSOR WILL CHECK THIS LINE
359
360 5$:
361 .ENDC
362 .ENDC
363
364 CALL @DDMSN ; CALL DDM TO SENSE MODEM STATUS
365 MOVB R4,M.CSV(R5) ; NEW CURRENT STATUS
366 TSTB M.TIM(R5) ; IS THERE AN ACTIVE TIMER?
367 BEQ 20$; NO ... DO NOTHING
368 DECB M.TIM(R5) ; YES ... REDUCE IT
369 BEQ 30$; IF NOW ZERO ... FORCE DISPATCH
370 CMPB M.CSV(R5),M.PSV(R5) ; HAS THE MODEM STATUS CHANGED?
371 BEQ 40$; NO ... DO NOTHING
372 MOVB M.LIN(R5),R4 ; GET SYSTEM LINE NUMBER
373
374 10$:
375 20$:
376 30$:
377
378 000310 105715
379 000312 100454
380 000314 105765 000003
381 000320 001446
382 000322 132715 000002
383 000326 001415
384 000330 116503 000001
385
386 000334 010302
387 000336 006302
388 000340 067702 000000G
389 000344 011202
390
391 000346 005712
392 000350 100032
393
394 000352 110465 000004
395 000356 105765 000002
396 000360 001403
397 000370 105365 000002
398 000374 001404
399 000376 126565 000004 000005
400 000400 001414
401 000406 116504 000001

```

```

673 .SBTTL SCHEDULE DLC PROCESS
674 ;+
675 **-$SCHPR-SCHEDULE DLC PROCESS
676 **-$SCHP2-SCHEDULE DLC PROCESS WITH GIVEN CCB (ALTERNATE ENTRY)
677 ;
678 AN EVENT HAS OCCURED WHICH REQUIRES THAT THE DLC PROCESS BE NOTIFIED.
679 ;
680 INPUTS:
681 R1 - SUBFUNCTION CODE
682 R2 - NEW LINE STATE
683 R3 - OPERATION COMPLETION STATUS
684 R4 - CCB TO USE (ALTERNATE ENTRY AT $SCHP2)
685 ;
686 OUTPUTS:
687 'C' CLEAR - DLC PROCESS SUCCESSFULLY POSTED
688 'C' SET - CCB ALLOCATION FAILURE
689 ;
690 .ENABL LSB
691
692 001100 $SCHPR::CALL @CCBG7 ; GRAB A CCB
693 001104 103415 BCS 10$; ALLOCATION FAILURE?
694 001106 110265 000003 $SCHP2::MOVB R2,M.S(T(R5) ; SET NEW LINE STATE
695 001112 010164 000010 MOV R1,C.FNC(R4) ; SET UP SUBFUNCTION CODE IN CCB
696 001116 116564 000001 000006 MOVB M.LIN(R5),C.LIN(R4) ; SET UP SYSTEM LINE NUMBER
697 001124 116564 000006 000007 MOVB M.STA(R5),C.STA(R4) ; AND STATION ADDRESS
698 001132 CALL @DDCCP ; POST COMPLETION TO DLC
699 001136 000241 CLC ; INDICATE SUCCESS
700 001140 10$: RETURN
701
702 .DSABL LSB
703 ;
704 .END
000001

```

```

128 MOV CMFRK,R4 ; NO ... POINT TO FORK BLOCK
129 CLR (R4) ; CLEAR OUT THE FORK BLOCK LINK WORD
130 MOV R2,-2(R4) ; STUFF UNIBUS RUN MASK IN FORK BLOCK
131 CALL @FORK ; AND RE-QUEUE IT
132 BR 55$; NOW EXIT THE SOFTWARE LEVEL
133
134 7$: .ENDC
135 .ENDC
136
137 000040 010402 9$: MOV R4,R2 ; COPY ADDRESS OF CCB
138 000042 032762 040000 000012 10$: BIT #CS.LST,C.STS(R2) ; LAST CCB OF A GROUP ?
139 000050 001002 20$: BNE 20$; IF NE YES
140 000052 011202 MOV (R2),R2 ; GET ADDRESS OF NEXT CCB
141 000054 000772 BR 10$; TEST AGAIN
142 000056 011213 20$: MOV (R2),(R3) ; SET NEW FIRST CCB ADDRESS
143 000060 001002 BNE 30$; IF NE, NO NEW LAST
144 000062 010363 000002 MOV R3,2(R3) ; CLOSE UP THE LIST
145 000066 30$: .IF DF R$$MPL
146 .IF NDF R$$PRO
147
148 BIT #F2.MP,@FMSK2 ; IS THIS A MULTIPROCESSOR?
149 BEQ 32$; BR IF NO
150 CALL @C(SP)+ ; CO-ROUTINE RETURN TO UNLOCK PROCESS QUEUE
151
152 32$: .ENDC
153 .ENDC
154
155 156 000066 MTPS #0 ; ENABLE INTERRUPTS
157 000072 005012 CLR (R2) ; CLEAR POINTER IN LAST CCB
158 000074 042762 040000 000012 35$: BIC #CS.LST,C.STS(R2) ; CLEAR MARKER BIT
159 000102 016402 000006 MOV C.LIN(R4),R2 ; TEST FOR PROCESS LAYER
160 000106 100030 BPL 45$; IF PL, DISPATCH TO A DLC
161
162 ; DISPATCH AN LLC PROCESS
163
164
165
166 000110 105764 000003 40$: TSTB C.BID(R4) ; MESSAGE FROM DLC OR LLC ?
167 000114 103403 BMI 42$; IF M1, FROM LLC - LEAVE C.LIN ALONE
168 000116 042764 100000 000006 42$: BIC #100000,C.LIN(R4) ; MARK MESSAGE FROM A DLC
169 000124 000302 SWAB R2 ; EXTRACT PDV INDEX
170 000126 142764 000200 000003 BICB #200,C.BID(R4) ; CLEAR LLC DESTINATION INDICATOR
171 000134 042702 177650 BIC #C<17>,R2 ; REMOVE CHANNEL AND LLC LAYER FLAG
172 000140 010403 MOV R4,R3 ; COMPUTE A POINTER TO THE FUNCTION CODE
173 000142 062703 000010 ADD #C.FNC,R3
174 000146 010205 MOV R2,R5 ; COPY PDV INDEX
175 000150 067705 000000G ADD @PDVTA,R5 ; POINT INTO PDV INDEX TABLE
176 000154 011505 MOV (R5),R5 ; GET PDV ADDRESS
177 000156 016505 000016 MOV Z.DAT(R5),R5 ; GET ADDRESS OF LLC'S DATA BASE DESCRIPTOR BLOCK
178 000162 CALL @PDDSP ; DISPATCH THE PROCESS
179 000166 000707 BR 5$; GO EXAMINE QUEUE AGAIN
180
181 ; DISPATCH A DLC PROCESS
182
183
184

```



AXSUB MACRO V05.03b Friday 28-Jun-85 18:30 Page 8  
\$ALOCX - Allocate core buffer in extended pool

```

172 .SBTTL $ALOCX - Allocate core buffer in extended pool
173
174 *-$ALOCX-Allocate core buffer in extended pool
175
176 This routine is called to allocate an core buffer in extended pool.
177 the allocation algorithm is first fit and blocks are allocated in
178 multiples of four bytes. If the extended allocation fails, the
179 block is allocated using sx pool.
180
181 Inputs:
182
183 R1=size of the core buffer to allocate in bytes.
184
185 Outputs:
186
187 c=1 if insufficient core is available to allocate the block.
188 c=0 if the block is allocated.
189 r0=address of the allocated block.
190 r1=length of block allocated
191
192 Registers:
193
194 R2 is modified
195
196
197 000020 $ALOCX::
198 .IF NDF M$$MGE
199
200 CALLR @ALOCB ; allocate from rsx pool in unmapped systems
201
202 .IFF
203
204 000020 010346 MOV R3,-(SP) ; save register
205 000022 SAVMAP ; save current apr 6 mapping
206 000026 012700 000000' MOV #XAVLL-2,R0 ; point to allocation mask word
207 000032 061001 ADD (R0),R1 ; add rounding factor
208 000034 042001 BIC (R0)+,R1 ; clear excess and point to list head
209 000036 017710 000000G MOV @XAVL,(R0) ; copy cex pool listhead
210 000042 010002 MOV R0,R2 ; initialize current unmapped pointer (r2)
211
212 000044 010203 10$: MOV R2,R3 ; move current unmapped ptr to prev unm ptr
213 000046 011002 MOV (R0),R2 ; get unmapped address of next block (current)
214 000050 001003 BNE 15$; if ne, another block to check
215 000052 CALL @ALOCB ; else, try to allocate from rsx pool
216 000056 000425 BR 30$; and return
217
218 000060 011046 15$: MOV (R0),-(SP) ; set up address for $ceacc (current ptr)
219 000062 CALL @CEACC ; map extended block
220 000066 012600 MOV (SP)+,R0 ; retrieve mapped address
221 000070 026001 000002 CMP 2(R0),R1 ; block big enough?
222 000074 103763 BLO 10$; if lo no
223 000076 001406 BEQ 20$; if eq block is exact size
224 000100 160160 000002 SUB R1,2(R0) ; calculate size remaining
225 000104 066002 000002 ADD 2(R0),R2 ; add size of remaining block to unm addr
226 000110 010200 MOV R2,R0 ; move retruned address to r0
227 000112 000407 BR 30$; and return
228

```

AXSUB      CREATED BY    MACRO    ON 28-JUN-85 AT 18:31      PAGE 3      E 6

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES |        |        |        |        |       |        |        |        |        |
|------------|------------|--------|--------|--------|--------|-------|--------|--------|--------|--------|
| CALL       | 8-215      | 8-219  | 8-231  | 9-281  | 9-293  | 9-310 | 10-380 | 10-392 | 11-460 | 12-493 |
|            | 12-496     | 12-522 |        |        |        |       |        |        |        |        |
| CALLR      | #5-59      | 9-275  |        |        |        |       |        |        |        |        |
| CCBD\$     | #5-57      | 5-61   |        |        |        |       |        |        |        |        |
| ENABL\$    | #5-56      |        |        |        |        |       |        |        |        |        |
| INHIB\$    | #5-56      |        |        |        |        |       |        |        |        |        |
| MAP        | #5-58      | 9-301  | 9-314  | 9-324  | 9-326  | 9-330 | 12-511 |        |        |        |
| PDVDF\$    | #5-57      | 5-62   |        |        |        |       |        |        |        |        |
| RESMAP     | #5-58      | 8-235  | 9-303  | 9-316  | 9-337  |       |        |        |        |        |
| RESRG      | #5-56      |        |        |        |        |       |        |        |        |        |
| RETURN     | 8-238      | 9-338  | 10-419 | 11-462 | 12-532 |       |        |        |        |        |
| SAVMAP     | #5-58      | 8-205  | 9-276  | 9-283  | 9-300  | 9-308 | 9-323  |        |        |        |
| SAVRG      | #5-55      |        |        |        |        |       |        |        |        |        |
| SOB        | 12-519     |        |        |        |        |       |        |        |        |        |

```

388
389 000370 132701 000004 BITB #XF.LLC,R1 ; LLC TIMER?
390 000374 001413 BEQ 30$; IF EQ, NO
391
392 000376 010205 MOV R2,R5 ; COPY PDV INDEX
393 000400 067705 ADD @PDVTA,R5 ; POINT INTO PDV INDEX TABLE
394 000404 011505 MOV (R5),R5 ; GET ADDRESS OF PROCESS' PDV
395 000406 016505 MOV Z.DAT(R5),R5 ; GET ADDRESS OF LLC DATABASE DESCRIPTOR
396 000412 016703 MOV STMFC,R3 ; SET UP FUNCTION CODE
397 000416 CALL @PDDSP ; AND DISPATCH TO THE PROCESS
398 000422 000417 BR 70$; ENTER COMMON CODE
399
400 000424 132701 000002 30$: BITB #XF.DLC,R1 ; DLC TIMER?
401 000430 001403 BEQ 40$; IF EQ, NO
402
403 000432 CALL @STD:1 ; SET UP DLC PDV INDEX AND LINE TABLE ADDRESS
404 000436 000402 BR 50$; AND ENTER COMMON CODE
405
406 000440 40$: CALL @STDD1 ; SET UP DDM PDV INDEX AND LINE TABLE ADDRESS
407 000444 016703 000000G 50$: MOV STMFC,R3 ; SET UP FUNCTION CODE
408 000450 CALL @PDSPL ; DISPATCH TO PROCESS
409 000454 000402 BR 70$; ENTER COMMON CODE
410
411 000456 60$: CALL STALT ; PUT CELL BACK ON TIMER QUEUE
412 000462 70$: RESRG <R0> ; RECOVER LISTHEAD ADDRESS
413 000464 012604 MOV (SP)+,R4 ; GET ADDRESS OF NEXT TIMER CELL
414 000466 001316 BNE 10$; IF NE, MORE TO GO
415
416 000470 MTPS #0 ; RESET PRIORITY
417 000474 100$: RETURN

```

SESCON - Session control connec MACRO V05.03b Friday 28-Jun-85 19:53 Page 7  
Connect and connect accept QIO processing

```

63 .SBTTL Connect and connect accept QIO processing
64 ;+
65 ***-SCNQIO-Connect and connect accept QIO processing
66 This routine processes all connect and connect accept requests.
67 -
68 Inputs:
69 R1 = Address of the task's header
70 R2 = I/O subfunction code/4
71 R3 = Address of I/O packet
72 R4 = Address of window block
73 R5 = Address of database descriptor
74 :
75 :
76 000000 016400 000012 0000000G $CNQIO: MOV W.MBOX(R4),R0 ; Set up the long term link timer
77 000004 016067 000016 0000000G MOV M.RESP(R0),$LTM ; ...
78 000012 010067 000000G MOV R0,$MAIBX ; Store the mail box address
79 000016 010467 000000G MOV R4,$WBLK ; Save address of window block
80
81 .IF DF N$$SLI
82 CLRB $LTYPE ; Indicate QIO interface for this link
83
84 .ENDC
85
86 000022 112767 000200 0000000G MOVB #LF.MMF,$FLOW ; Assume message flow control
87 000030 105763 000000G TSTB I.FCN(R3) ; Is this a non-flow controlled link?
88 000034 100002 000000G BPL 10$; If PL, no
89 000036 105067 000000G CLRB $FLOW ; Set up for no flow-control
90 000042 000000 000000G CALL @CONTBL(R2) ; Dispatch to processing routine
91 000046 016703 000000G MOV $WBLK,R3 ; Recover window block address
92 000052 103413 000000G BCS 20$; If CS, request failed
93
94 000054 000000 000000G MAPLLT ; Recover mapping to the LLT
95 000062 016501 000026 000000G MOV N$LLT(R5),R1 ; Recover virtual address of LLT
96 000066 010361 000040 000000G MOV R3,L.WIND(R1) ; Set up window address in LLT
97 000072 016563 000030 000004G MOV N$PLLT(R5),W.LLT(R3)
98 000100 000000 000000G RETURN
99
100 000102 000000 000000G 20$: CALLR RMVWND ; Release the window block resources
101
102 ;+
103 ; Connect function dispatch table
104 :-
105 000000 000000 000000G .PSECT $HIGH
106
107 000000 000106' 000000G CONTBL: .WORD CON ; Connect
108 000002 000300' 000000G .WORD ACC ; Accept
109
110 000106 000000 000000G .PSECT

```

SESCON - Session control connec MACRO V05.03b Friday 28-Jun-85 19:53 Page 8

SESCON11S CREATED BY MACRO ON 28-JUN-85 AT 19:53 PAGE 3 E 9

MACRO CROSS REFERENCE CREF 04.00

| MACRO NAME | REFERENCES |        |        |        |        |       |       |        |        |        |
|------------|------------|--------|--------|--------|--------|-------|-------|--------|--------|--------|
| CALL       | 7-91       | 8-152  | 8-154  | 8-159  | 9-186  | 9-192 | 9-196 | 10-249 | 10-252 | 10-257 |
|            | 10-310     | 11-359 | 11-412 | 11-415 |        |       |       |        |        |        |
| CALLR      | 7-101      | 8-167  |        |        |        |       |       |        |        |        |
| CCBDF\$    | #6-47      | 6-38   |        |        |        |       |       |        |        |        |
| CNBDF\$    | #6-47      | 6-57   |        |        |        |       |       |        |        |        |
| CRBDF\$    | #6-46      | 6-50   |        |        |        |       |       |        |        |        |
| DHBDF\$    | #6-45      | 6-49   |        |        |        |       |       |        |        |        |
| ECDD8\$    | #6-46      | 6-53   |        |        |        |       |       |        |        |        |
| LLTDF\$    | #6-46      | 6-54   |        |        |        |       |       |        |        |        |
| LLWDF\$    | #6-46      | 6-51   |        |        |        |       |       |        |        |        |
| MAP        | #6-44      | 7-95   | 8-132  |        |        |       |       |        |        |        |
| MAPLLT     | #6-44      | 7-95   |        |        |        |       |       |        |        |        |
| MBXDF\$    | #6-46      | 6-52   |        |        |        |       |       |        |        |        |
| NSFDF\$    | #6-47      | 6-56   |        |        |        |       |       |        |        |        |
| NSSYM\$    | #6-47      | 6-59   |        |        |        |       |       |        |        |        |
| OBJDF\$    | #6-46      | 6-55   |        |        |        |       |       |        |        |        |
| RECMAP     | #6-44      | 8-144  |        |        |        |       |       |        |        |        |
| RESRG      | #6-44      | 10-330 |        |        |        |       |       |        |        |        |
| RETURN     | 7-99       | 8-163  | 9-198  | 10-331 | 11-419 |       |       |        |        |        |
| SAVRG      | #6-44      | 10-223 |        |        |        |       |       |        |        |        |
| SOB        | 8-137      | 8-142  |        |        |        |       |       |        |        |        |
| \$IERRC    | #6-45      | 9-200  | 9-202  |        |        |       |       |        |        |        |

SESCIL - Session control contro MACRO V05.03b Friday 28-Jun-85 19:53 <sup>E 10</sup> Page 11-1  
Get network data

|     |        |         |       |     |   |    |                        |
|-----|--------|---------|-------|-----|---|----|------------------------|
| 387 | 000434 | 000550' | .WORD | DSC | : | 3  | - user disconnect      |
| 388 | 000436 | 000550' | .WORD | ABT | : | 4  | - user abort           |
| 389 | 000440 | 000540' | .WORD | ABO | : | 5  | - network disconnect   |
| 390 | 000442 | 000576' | .WORD | EVT | : | 6  | - network event        |
| 391 | 000444 | 000446' | .WORD | VFY | : | 7  | - network verification |
| 392 |        |         | .WORD | MNP | : | 10 | - mop event            |

|                  |                 |                  |                   |                   |
|------------------|-----------------|------------------|-------------------|-------------------|
| LA.ACK= 100000   | L.MASZ 000072   | M.RESP 000016    | NT\$DAT= 000002   | N.CBL = 000142    |
| LA.CRS= 020000   | L.NIN 000020    | M.SPA 000012     | NT\$DC = 000012   | N.CDA 000142      |
| LA.MSK= 170000   | L.NXN 000016    | M.TASK 000004    | NT\$DIS= 000014   | N.CDAC 000140     |
| LA.NAK= 110000   | L.NXTH 000010   | M.USE 000010     | NT\$DLS= 000006   | N.CDDS 000070     |
| LA.NMS= 010000   | L.OPD 000103    | NC.FM0= 000000   | NT\$ILS= 000010   | N.CDEV 000062     |
| LA.RES= 040000   | L.OPDL 000102   | NC.FM1= 000001   | NT\$IMS= 000002   | N.CID 000064      |
| LA.WND= 004000   | L.REM 000006    | NC.FM2= 000002   | NT\$INT= 000004   | N.CIDC 000062     |
| LDBGT = ***** GX | L.RFC 000050    | NE\$ABM= 000010  | NT\$RET= 000032   | N.CPS 000106      |
| LDBRT = ***** GX | L.RLA 000004    | NE\$ABO= 000046  | NT\$ROU= 000024   | N.CPSC 000104     |
| LD\$LP = 000000  | L.RNC 000022    | NE\$ACC= 000042  | NT\$RTR= 000030   | N.CTL 000000      |
| LF.DRD= 000004   | L.RTU 000060    | NE\$ACT= 000042  | NT\$TSP= 000026   | N.CUIC 000066     |
| LF.FRC= 000001   | L.RTYD 000055   | NE\$COM= 000047  | NT.ABO= 000005    | N.CUNI 000064     |
| LF.HFQ= 000010   | L.RTYI 000057   | NE\$CFM= 000050  | NT.ABT= 000004    | N.DDE 000010      |
| LF.HMF= 000040   | L.SEC 000064    | NE\$FMT= 000005  | NT.CON= 000001    | N.DDEC 000006     |
| LF.HSF= 000020   | L.SEGZ 000076   | NE\$GEN= 000007  | NT.DSC= 000003    | N.DFM 000004      |
| LF.IRD= 000002   | L.STA 000000    | NE\$IFC= 000030  | NT.EVT= 000006    | N.DGP 000006      |
| LF.MMF= 000200   | L.TC 000042     | NE\$ILS= 000043  | NT.INT= 000002    | N.DNM 000014      |
| LF.MSF= 000100   | L.TIC 000043    | NE\$IMG= 000042  | NT.MOP= 000010    | N.DMMC 000012     |
| LS.DLS= 100000   | L.TIPD 000013   | NE\$MLB= 000006  | NT.NSP= 000010    | N.DOT 000005      |
| LS.FCC= 000004   | L.TIPI 000012   | NE\$NNF= 000012  | NT.VFY= 000007    | N.DUS 000010      |
| LS.FCO= 000001   | L.TMRD 000054   | NE\$NOD= 000002  | NVP 001006R       | N.SDE 000042      |
| LS.FCT= 000002   | L.TMRI 000056   | NE\$NSD= 000003  | NSACQ 000000      | N.SDEC 000040     |
| LS.ILS= 100000   | L.TYP 000001    | NE\$NSL= 000013  | NSACTL 000032     | N.SEGZ 000002     |
| LS.MAK= 000020   | L.USA 000024    | NE\$NSR= 000003  | NSCIR 000034      | N.SFM 000036      |
| LS.MNK= 000040   | L.USTA 000036   | NE\$RES= 000001  | NSDLA 000020      | N.SGP 000040      |
| LS.RES= 000360   | L.VER 000015    | NE\$SSR= 000000  | NSDLY 000014      | N.SND 000030      |
| LS.RSV= 000300   | L.WIND 000040   | NE\$SSS= 000045  | NSLEN 000054      | N.SNM 000046      |
| LT.CCA= 000020   | MAPBF 000662R   | NE\$TCN= 000040  | NSENC 000042      | N.SMMC 000044     |
| LT.DIR= 000010   | MA.CI = 000040  | NE\$TCO= 000006  | NSERRC 000022     | N.SOT 000037      |
| LT.LCL= 000001   | MA.DA = 000000  | NE\$TPA= 000010  | NSFLG 000005      | N.SUS 000042      |
| LT.LPL= 000002   | MA.IL = 000020  | NE\$UDB= 000004  | NSFNC 000006      | OPN 000024R 002   |
| LT.NOT= 000040   | MC.CC = 000040  | NE\$UBLK= 000100 | NSGENQ 000052     | PEM 000350R 002   |
| LT.RSU= 000200   | MC.CI = 000020  | NE\$DMO= 000010  | NSGTM 000015      | PS\$P45= 000000   |
| LT.SLI= 000004   | MC.DC = 000100  | NE\$MOU= 000040  | NSHIGH 000033     | PS\$WRD= 000000   |
| LT.TDA= 000100   | MC.DI = 000060  | NE\$RST= 000002  | NSLLT 000026      | QUETSK= ***** GX  |
| LS\$ASG= 000000  | MC.ND = 000000  | NE\$SCN= 000020  | NSLLTM 000024     | Q\$SOP= 000010    |
| LS\$DRV= 000000  | MC.RC = 000140  | NE\$SHU= 000004  | NSLVC 000036      | RDBSZ = ***** GX  |
| LS\$P11= 000001  | MD.BM = 000040  | NE\$TIM= 000200  | NSMBXQ 000050     | REJECT = ***** GX |
| LS\$11R= 000000  | MD.EM = 000100  | NMSARA= 176000   | NSPLLT 000030     | RETRES= ***** GX  |
| L.CSTA 000037    | MD.ILS= 000040  | NMSWOD= 001777   | NSSLA 000016      | RMVWND= ***** GX  |
| L.CTR 000074     | MD.IM = 000020  | NO.DTR= 000077   | NS\$NDD 000012    | R\$SDER= 000000   |
| L.DCR 000100     | MF.ACK= 000004  | NO.FAL= 000021   | NS\$TIM 000004    | R\$SK11= 000001   |
| L.FLAG 000014    | MF.CTL= 000010  | NO.FA1= 000001   | NSVCB 000010      | R\$SND= 000000    |
| L.LLSQ 000052    | MF.DAT= 000000  | NO.NCU= 000023   | NS\$ACC= 000001   | R\$S11M= 000000   |
| L.ILTT 000066    | MOV DAT 000616R | NO.RTL= 000022   | NS\$ACK= 000011   | R\$S11S= 000000   |
| L.LDA 000032     | MS\$CRB= 000124 | NO.TAS= 000000   | NS\$EVL= 000001   | SPA 000172R 002   |
| L.LIA 000034     | MS\$CRX= 000000 | NO.TCL= 000017   | NS\$HDR= 000007   | SRSTD = ***** GX  |
| L.LLA 000002     | MS\$FCS= 000000 | NO.TCT= 000005   | NS\$LDV= 000001   | ST\$CC = 000004   |
| L.LNG 000124     | MS\$MGE= 000000 | NO.TLK= 000020   | NS\$MLL= 000001   | ST\$CIR= 000006   |
| L.LNO 000026     | MS\$MUP= 000000 | NS\$DON= 000000  | NS\$MOV= 000010   | ST\$CIS= 000002   |
| L.LPT 000065     | MS\$NET= 000000 | NS\$SDI= 000002  | NS\$NCT= 000001   | ST\$DAT= 000010   |
| L.LSA 000030     | MS\$OVR= 000000 | NS\$WDC= 000004  | NS\$OVR= 000022   | ST\$DIP= 000012   |
| L.LSFD 000046    | M.MAIL 000014   | NT\$AKD= 000020  | NS\$PEM= 000001   | ST\$PND= 000014   |
| L.LSFI 000044    | M.MAX 000011    | NT\$AKI= 000022  | NS\$SES= 000001   | S\$SWRG= 000000   |
| L.LTT 000062     | M.MBL 000020    | NT\$CC = 000016  | NS\$SMC= ***** GX | S\$YSZ= 007600    |
| L.MASQ 000070    | M.NAST 000007   | NT\$CON= 000000  | N.CAC 000120      | TDISP 000430R     |
| L.MAST 000073    | M.NEXT 000002   | NT\$CTL= 000000  | N.CACC 000116     | TLACHK= ***** GX  |

|                  |                 |                 |                  |                  |
|------------------|-----------------|-----------------|------------------|------------------|
| AC\$DNT= 000002  | CL\$XL2= 013700 | C.BUF2 000024   | ER\$STA= 000051  | EV.LCB= 000100   |
| AC\$X25= 000001  | CL\$XL3= 013600 | C.CNT 000020    | ER\$UOB= 000004  | EV.LIN= 000004   |
| AE\$CIR= 000003  | CL\$X2S= 013500 | C.CNT1 000020   | EVDSC = ***** GX | EV.MAP= 000002   |
| AE\$LIN= 000001  | CL.MU1= 000001  | C.CNT2 000030   | EVLSES 000150RG  | EV.MOD= 000040   |
| AE\$MOD= 000004  | CL.MU2= 000002  | C.FLG 000022    | EVL\$ACF= 000201 | EV.MOD= 000010   |
| AS\$CHK= 000000  | CL.RES= 177774  | C.FLG1 000022   | EV\$ADR= 000420  | EV.PRT= 000200   |
| AS\$CPS= 000000  | CM.CIR= 000002  | C.FLG2 000032   | EV\$ADU= 000417  | ES\$DATA 000020  |
| AS\$PRI= 000000  | CM.FMT= 100000  | C.FNC 000010    | EV\$APL= 000400  | ES\$EVT\$ 000000 |
| AS\$TRP= 000000  | CM.HRD= 000002  | C.LIN 000006    | FV\$ARC= 000421  | ES\$LCN 000016   |
| CB.CCB= 000002   | CM.LIN= 000000  | C.LNK 000000    | EV\$AUC= 000010  | ES\$LIN 000000   |
| CB.DDM= 000040   | CM.LOO= 000001  | C.MOD 000011    | EV\$AUS= 000003  | ES\$MOD 000012   |
| CB.DLC= 000020   | CM.XLO= 000004  | C.NSP 000004    | EV\$CDF= 000520  | ES\$NBR 000014   |
| CB.RDB= 000004   | CP.DCF= 000040  | C.PRO 000042    | EV\$COZ= 000011  | ES\$NBS 000020   |
| CB.SDB= 000010   | CP.HDL= 000007  | C.RSV 000002    | EV\$CDB= 000302  | ES\$NCR 000034   |
| CB.SLI= 000100   | CP.PS = 177400  | C.STA 000007    | EV\$GAS= 035101  | ES\$NCS 000036   |
| CB.XLB= 000001   | CP.PS1= 000200  | C.STS 000012    | EV\$HCE= 035114  | ES\$NIC 000044   |
| CC.LLC= 000200   | CP.XCF= 000100  | C.URM 177776    | EV\$HCI= 035113  | ES\$NLEN 000050  |
| CELOG = ***** GX | CP.2FR= 000030  | C.XACP 000004   | EV\$HFE= 000506  | ES\$NLLA 000012  |
| CE.ABO= 100362   | CS.ABO= 000100  | C.XID 000035    | EV\$IFL= 000413  | ES\$NLNK 000000  |
| CE.DAO= 100346   | CS.BRO= 000002  | C.XLEN 000044   | EV\$IFO= 000415  | ES\$NML 000040   |
| CE.DIS= 100366   | CS.BUF= 000200  | C.XPL1 000040   | EV\$IFS= 000414  | ES\$NMR 000024   |
| CE.ERR= 100370   | CS.CES= 000002  | C.XPT 000034    | EV\$INF= 000515  | ES\$NMS 000030   |
| CE.ILN= 100350   | CS.CHN= 000010  | C.XSVC 000042   | EV\$LDL= 000407  | ES\$NOD 000002   |
| CE.LTO= 100356   | CS.CMP= 000200  | C.XTC 000037    | EV\$LDN= 010416  | ES\$NOD 000010   |
| CE.MOP= 100372   | CS.DCR= 000400  | C.X25 000036    | EV\$LDO= 000411  | ES\$NRT 000042   |
| CE.NTE= 100361   | CS.DEF= 000004  | DL\$AST= 000302 | EV\$LDS= 000410  | ES\$NRP 000005   |
| CE.RTE= 100376   | CS.DEV= 000002  | DL\$HLT= 000000 | EV\$LSC= 000500  | ES\$NSEG 000010  |
| CE.SRC= 100364   | CS.DIS= 000040  | DL\$IST= 000001 | EV\$LUP= 000412  | ES\$NTIM 000046  |
| CE.STP= 100352   | CS.ENA= 000001  | DL\$MAI= 000004 | EV\$NOL= 000402  | ES\$NUSE 000004  |
| CE.TME= 100354   | CS.ENB= 000020  | DL\$OFF= 000001 | EV\$NRC= 000416  | ES\$PORT 000014  |
| CE.TMO= 100374   | CS.ERR= 100000  | DL\$ON = 000000 | EV\$NSC= 000200  | ES\$PRM 000002   |
| CE.UNS= 100344   | CS.FTL= 010100  | DL\$RUN= 000003 | EV\$NUL= 000401  | ES\$STAT 000006  |
| CF.CHN= 000001   | CS.HCR= 000001  | DL\$SHU= 000002 | EV\$NVR= 000406  | ES\$STR 000006   |
| CF.EOM= 000004   | CS.HFE= 002000  | DL\$SYN= 000005 | EV\$OPL= 000403  | ESTCB 000004     |
| CF.HDR= 000020   | CS.LST= 040000  | D\$BLG= 177514  | EV\$PCC= 034000  | ES\$XPR= 000000  |
| CF.LB = 100000   | CS.MTL= 004000  | D\$ISK= 000000  | EV\$PCI= 034001  | E.CTL 000020     |
| CF.LIN= 000002   | CS.RNG= 000010  | D\$LL1= 000001  | EV\$PCM= 034002  | E.DATA 000046    |
| CF.SOM= 000010   | CS.ROV= 000004  | D\$SYNC= 000000 | EV\$PFE= 000404  | E.EVT 000002     |
| CF.SYN= 000040   | CS.RSN= 010000  | D\$SYNM= 000000 | EV\$PPC= 034003  | E.LCN 000042     |
| CF.TRN= 000100   | CS.SHU= 000001  | EF\$ACT= 000001 | EV\$RCF= 000517  | E.LEN 000216     |
| CL\$ASZ= 010503  | CS.SID= 000002  | ENCCIR 000134R  | EV\$RDC= 010001  | E.LIN 000024     |
| CL\$DLL= 000500  | CS.STR= 000004  | ENCSR 000114R   | EV\$RDR= 010002  | E.LNK 000000     |
| CL\$ECL= 000300  | CS.SUC= 000001  | ENC1W 000076R   | EV\$RJE= 035106  | E.MOD 000036     |
| CL\$LDN= 010400  | CS.TMO= 020000  | ENC2W 000106R   | EV\$RSC= 000501  | E.NOD 000034     |
| CL\$MAN= 000000  | CS.XUR= 000004  | ERSABM= 000010  | EV\$RUL= 000405  | E.PDV 000021     |
| CL\$MFL= 000010  | CTRSES 000000RG | ERSABO= 000046  | EV\$SNA= 035000  | E.PORT 000040    |
| CL\$PAZ= 034100  | CTRIBL 000066R  | ERSABT= 000011  | EV\$SNF= 000516  | E.PRM 000026     |
| CL\$PLH= 034000  | CV\$MSK= 000003 | ERSACC= 000042  | EV\$SPE= 035001  | E.PVC 000044     |
| CL\$PLL= 000600  | CV\$31 = 000001 | ERSCDI= 000052  | EV\$XCE= 034110  | E.SIZ 000022     |
| CL\$PRT= 034200  | CV\$32 = 000000 | ERSCOM= 000047  | EV\$XDI= 013600  | E.TIME 000004    |
| CL\$ROU= 010000  | CV\$40 = 000002 | ERSFMT= 000005  | EV\$XGW= 034111  | FC.CCP= 000020   |
| CL\$SES= 000200  | CS\$ORE= 000400 | ERSMLB= 000006  | EV\$XMX= 000514  | FC.CTL= 000006   |
| CL\$SFL= 000004  | CS\$RSH= 177564 | ERSNNF= 000012  | EV\$XRS= 000512  | FC.KCP= 000016   |
| CL\$SGE= 035000  | C.ADD 000034    | ERSNOD= 000002  | EV\$XSC= 000513  | FC.KIL= 000004   |
| CL\$SSE= 035100  | C.BID 000003    | ERSNSI= 000013  | EV\$X2S= 013500  | FC.MAN= 000024   |
| CL\$TRN= 000400  | C.BUF 000014    | ERSNSR= 000003  | EV.CCB= 000001   | FC.MLD= 000026   |
| CL\$TPY= 000001  | C.BUF1 000014   | ERSRES= 000001  | EV.CIR= 000020   | FC.PCT= 000030   |



SESDAT - Session control local MACRO V05.03b Friday 28-Jun-85 19:54 Page 7

Local data

```

48 .SBTTL Local data
49
50 000000 000000 $SESDB::WORD 0 ; Address of ECL database descriptor
51 000002 000000 $SESPD::WORD 0 ; PDV index of ECL process
52
53 000004 000000 $UCB::WORD 0 ; Address of device's UCB
54
55 000006 000000 $RQTCB::WORD 0 ; ICB being requested
56 000010 000000 000000 $RQNAM::WORD 0,0 ; Requested task name
57
58 000014 000000 $RQCPY::WORD 0 ; Max # of copies allowed
59
60 000016 000000 $MAIBX::WORD 0 ; Current mailbox
61
62 000020 000000 $BYTE::WORD 0 ; Byte count
63
64 000022 000000 $RCCB::WORD 0 ; Address of received CCB
65 000024 000 $SRVCS::BYTE 0 ; Services requested in a CI message
66 000025 000 $INFO::BYTE 0 ; Info field from CI message
67 000026 000000 $SEGMT::WORD 0 ; Segment size from CI message
68 000030 000000 $MENU::WORD 0 ; Menu field from CI message
69 000032 000000 $LADDR::WORD 0 ; Local area address for CI message
70
71 000034 000000 $REASN::WORD 0 ; Reason code from DI or DC message
72
73 000036 000000 $ENCOD::WORD 0 ; LLA encoding word
74
75 000040 000000 $WBLK::WORD 0 ; Address of current window block
76 000042 000000 $IOPKT::WORD 0 ; Address of current I/O packet
77
78 000044 000000 $LTM::WORD 0 ; Long term timer value
79 000046 000 $FLOW::BYTE 0 ; Flow control options
80
81 .IF DF N$$SLI
82
83 $LTYPE::BYTE 0 ; Link type flags
84
85 .ENDC
86
87 000047 000 $FLAGS::BYTE 0 ; General ACP flag word
88 ; Bit 0 = 'loopback node' flag
89 ;
90
91 .EVEN
92
93 .IF DF X$$HDR
94
95 $HDRMP::WORD 0 ; Task header mapping
96
97 .ENDC
98
99 .IF DF N$$ACC
100
101 000050 055400 131574 VFYNAM::RAD50 /NVP.../ ; Name of verification task
102
103 .ENDC
104

```

```

54 .SBITL Disconnect and connect reject QIO processing
55 ;+
56 **-$DSQIO-Disconnect and connect reject QIO processing
57 ;
58 This routine processes all disconnect and connect reject requests.
59 ;
60 Inputs:
61 R1 = Address of the task's header
62 R2 = I/O subfunction/4
63 R3 = Address of I/O packet
64 R4 = Address of window block/mailbox
65 R5 = Address of database descriptor
66 ;
67 .PSECT $HIGH
68 ;
69 $DSQIO::TST R4 ; Is there a window/mailbox ?
70 BEQ 10$; If EQ, no - just complete I/O
71 CALLR @DISTBL(R2) ; Else, dispatch to processing routine
72 ;
73 10$: CALLR IOSUC ; Complete I/O request
74 ;
75 ;+
76 ; Disconnect function dispatch table
77 ; -
78 ;
79 000014 000000' DISTBL: .WORD DSC ; Disconnect
80 000016 000016' .WORD ABT ; Abort
81 000020 000076' .WORD REJ ; Connect reject

```

|                  |                  |                 |                  |                |
|------------------|------------------|-----------------|------------------|----------------|
| ACCLLT= ***** GX | ERSABO= 000046   | EV\$SNF= 000516 | F\$SLVL= 000001  | L.ILSQ 000052  |
| AC\$DNT= 000002  | ERSABT= 000011   | EV\$SPE= 035001 | G\$STPP= 000000  | L.ILTT 000066  |
| AC\$X25= 000001  | ERSACC= 000042   | EV\$XCE= 034110 | G\$STSS= 000000  | L.LDA 000032   |
| AESCIR= 000003   | ERSCDI= 000052   | EV\$XDI= 013600 | G\$STTK= 000000  | L.LTA 000034   |
| AESLIN= 000001   | ERSCOM= 000047   | EV\$XGW= 034111 | G\$SWRD= 000000  | L.LLA 000002   |
| AESMGO= 000004   | ERSFMT= 000005   | EV\$XMX= 000514 | IN.DAT= 000400   | L.LNG 000124   |
| AS\$CHK= 000000  | ERSMLB= 000006   | EV\$XRS= 000512 | IN.ILS= 000001   | L.LNO 000026   |
| AS\$CPS= 000000  | ERSNNF= 000012   | EV\$XSC= 000513 | IOSUC= ***** GX  | L.LPT 000065   |
| AS\$PRI= 000000  | ERSNOD= 000002   | EV\$X2S= 013500 | IS\$RAR= 000000  | L.LSA 000030   |
| AS\$TRP= 000000  | ERSNSL= 000013   | EV\$XCB= 000001 | IS\$RDN= 000000  | L.LSFD 000046  |
| BYTE3= 000300    | ERSNSR= 000003   | EV.CIR= 000020  | I.FCN= ***** GX  | L.LSFI 000044  |
| CL\$ASZ= 010500  | ERSRES= 000001   | EV.LCB= 000100  | KILLNK= ***** GX | L.LTT 000062   |
| CL\$DLL= 000500  | ERSSTA= 000051   | EV.LIN= 000004  | K\$SCNT= 177546  | L.MASQ 000070  |
| CL\$ECL= 000300  | ERSUOB= 000004   | EV.MAP= 000002  | K\$SCSR= 177546  | L.MAST 000073  |
| CL\$LDN= 010400  | EVLSES= ***** GX | EV.MOD= 000040  | K\$SLDC= 000000  | L.MASZ 000072  |
| CL\$MAN= 000000  | EV\$ACF= 000201  | EV.NOD= 000010  | K\$STPS= 000074  | L.NIN 000020   |
| CL\$MFI= 000010  | EV\$ADR= 000420  | EV.PRT= 000200  | LA.ACK= 100000   | L.NXN 000016   |
| CL\$PAZ= 034100  | EV\$ADU= 000417  | ESDATA 000020   | LA.CRS= 020000   | L.NXTH 000010  |
| CL\$PLH= 034000  | EV\$APL= 000400  | ES\$EVS 000000  | LA.MC= 170000    | L.OPD 000103   |
| CL\$PLI= 000600  | EV\$ARC= 000421  | ESLNC 000016    | LA.NAK= 110000   | L.OPDL 000102  |
| CL\$PRT= 034200  | EV\$AUC= 000010  | ESLIN 000000    | LA.NMS= 010000   | L.REM 000006   |
| CL\$RCU= 010000  | EV\$AUS= 000003  | ESMOD 000012    | LA.RES= 040000   | L.RFC 000050   |
| CL\$SES= 000200  | EV\$CDF= 000520  | ENOD 000010     | LA.WND= 004000   | L.RLA 000004   |
| CL\$SFI= 000004  | EV\$COZ= 000011  | ESPORT 000014   | LD\$LP= 000000   | L.RNO 000022   |
| CL\$SGE= 035000  | EV\$DBR= 000302  | ESPRM 000002    | LF.DRD= 000004   | L.RTO 000060   |
| CL\$SSE= 035100  | EV\$GAS= 035101  | ESSTAT 000006   | LF.FRC= 000001   | L.RTYD 000055  |
| CL\$TRN= 000400  | EV\$HCE= 035114  | ESTCB 000004    | LF.HFO= 000010   | L.RTYI 000057  |
| CL\$TYP= 000001  | EV\$HCI= 035113  | ES\$XPR= 000000 | LF.HMF= 000040   | L.SEC 000064   |
| CL\$XL2= 013700  | EV\$HFE= 000506  | E.CTL 000020    | LF.HSF= 000020   | L.SEGZ 000076  |
| CL\$XL3= 013600  | EV\$IFL= 000413  | EV.DATA 000046  | LF.IRD= 000002   | L.STA 000000   |
| CL\$X2S= 013500  | EV\$IFO= 000415  | EV.EVT 000002   | LF.MMF= 000200   | L.TC 000042    |
| CL.MU1= 000001   | EV\$IFS= 000414  | E.LCN 000042    | LF.MSF= 000100   | L.TIC 000043   |
| CL.MU2= 000002   | EV\$INF= 000515  | E.LEN 000216    | LS.DLS= 100000   | L.TIPD 000013  |
| CL.RES= 177774   | EV\$LDL= 000407  | E.LIN 000024    | LS.FCC= 000004   | L.TIPI 000012  |
| CV\$MSK= 000003  | EV\$LDN= 010416  | E.LNK 000000    | LS.FCO= 000001   | L.TMRD 000054  |
| CV\$31= 000001   | EV\$LDO= 000411  | E.MOD 000036    | LS.FCI= 000002   | L.TMRI 000056  |
| CV\$32= 000000   | EV\$LDS= 000410  | E.NOD 000034    | LS.ILS= 100000   | L.TYP 000001   |
| CV\$40= 000002   | EV\$LSC= 000500  | E.PDV 000021    | LS.MAK= 000020   | L.USA 000024   |
| C\$SORE= 000400  | EV\$LUP= 000412  | E.PORT 000040   | LS.MNK= 000040   | L.USTA 000036  |
| C\$SRSH= 177564  | EV\$NOL= 000402  | E.PRM 000026    | LS.RES= 000360   | L.VER 000015   |
| DL\$AST= 000002  | EV\$NRC= 000416  | E.PVC 000044    | LS.RSV= 000300   | L.WIND 000040  |
| DL\$HLT= 000000  | EV\$NSC= 000200  | E.SIZ 000022    | LT.CCA= 000020   | MA.CI= 000040  |
| DL\$IST= 000001  | EV\$NUL= 000401  | E.TIME 000004   | LT.DIR= 000010   | MA.DA= 000000  |
| DL\$MAI= 000004  | EV\$NVR= 000406  | FLSLNK 000000R  | LT.LCL= 000001   | MA.IL= 000020  |
| DL\$OFF= 000001  | EV\$OPL= 000403  | FR\$BCC= 000007 | LT.LPL= 000002   | MC.CC= 000040  |
| DL\$ON= 000000   | EV\$PCC= 034000  | FR\$CCF= 000001 | LT.NOT= 000040   | MC.CI= 000020  |
| DL\$RUN= 000003  | EV\$PCI= 034001  | FR\$CDF= 000002 | LT.RSU= 000200   | MC.DC= 000100  |
| DL\$SHU= 000002  | EV\$PCM= 034002  | FR\$DAO= 000011 | LT.SLI= 000004   | MC.DI= 000060  |
| DL\$SYN= 000005  | EV\$PFE= 000404  | FR\$EXC= 000000 | LT.TDA= 000100   | MC.NO= 000000  |
| DMOTA 000062R    | EV\$PPC= 034003  | FR\$FRM= 000010 | L\$ASG= 000000   | MC.RC= 000140  |
| D\$SBUG= 177514  | EV\$RCF= 000517  | FR\$FTL= 000005 | L\$DRV= 000000   | MD.BM= 000040  |
| D\$SISK= 000000  | EV\$RDC= 010001  | FR\$OPN= 000004 | L\$P11= 000001   | MD.EM= 000100  |
| D\$SL11= 000001  | EV\$RDR= 010002  | FR\$RFD= 000006 | L\$11R= 000000   | MD.ILS= 000040 |
| D\$SYNC= 000000  | EV\$RJE= 035106  | FR\$SBU= 000012 | L.CSTA 000037    | MD.IM= 000020  |
| D\$SYNM= 000000  | EV\$RSC= 000501  | FR\$SHO= 000003 | L.CTR 000074     | MF.ACK= 000004 |
| EF\$ACT= 000001  | EV\$RUL= 000405  | FR\$SUB= 000013 | L.DCR 000100     | MF.CTL= 000010 |
| ER\$ABM= 000010  | EV\$SNA= 035000  | FR\$UPT= 000014 | L.FLAG 000014    | MF.DAT= 000000 |

```

98 .SBTTL Session control ACP idle loop
99
100 ;
101 ;**--ACPIDL-Session control ACP idle loop
102 ;
103 ; This routine is the main dispatch point for the session control
104 ; ACP. Process as many CCB's and/or I/O packets as possible before
105 ; returning to the caller.
106 ;
107 ; Inputs:
108 ; R5 = Address of database descriptor
109 ;
110 ; Outputs: (to ccb processing routines)
111 ; R3 = Subfunction code
112 ; R4 = Address of CCB
113 ; R5 = Address of database descriptor
114 ;
115 ; (To I/O packet processing routines)
116 ; R1 = Address of the task header
117 ; R2 = I/O subfunction code/4
118 ; R3 = Address of I/O packet
119 ; R4 = Address of window block or mailbox
120 ; R5 = Address of database descriptor
121 ; the task header will be mapped (RSX-11M-Plus only)
122 ;
123 .PSECT
124 ACPIDL::SWSTK$ RET ; Enter system state
125
126 000000 016777 000000G 000000G MOV $SESPD,ACMPDV ; Pretend we are ECL
127 000012 142765 000200 000005 BICB #NFTIM,NFLG(R5); Assume no more timer support required
128
129 000020 105765 000004 TSTB N$TIM(R5) ; Is the timer active?
130 000024 001412 BEQ 30$; If EQ, no
131
132 000026 105365 000004 10$: CALL SESTIM ; Perform session control timing
133 000032 001373 DECB N$TIM(R5) ; Reduce outstanding timer count
134 000036 BNE 10$; Loop till all processed
135
136 000040 011504 000054 20$: RECMAP ; Recover high APR mapping
137 000046 001424 CALL SCNGNQ ; Scan general delivery queue
138 000052 001424 MOV (R5),R4 ; Get address of next CCB to process
139 000054 BEQ 50$; If EQ, none
140
141 000056 011415 MOV (R4),(R5) ; Unlink CCB from chain
142 000060 001002 BNE 40$; If NE, more entries in the chain
143 000062 010565 MOV R5,2(R5) ; Update tail pointer
144
145 000066 152765 000200 000005 40$: BISB #NF$TIM,N$FLG(R5); Show we still need timer support
146 000074 005014 CLR (R4) ; Indicate this is a single CCB
147 000076 010467 000000G MOV R4,$RCCB ; Save address of CCB
148 000102 116402 000010 MOVB C.FNC(R4),R2 ; Get the function code
149 000106 116403 000011 MOVB C.MOD(R4),R3 ; and the modifier
150 000112 RECMAP ; Restore high APR mapping
151 000120 CALL @CCBDSP(R2) ; Dispatch to processing routine
152 000124 000752 BR 30$; then try for next CCB
153
154 000126 142765 000100 000005 50$: BICB #NF$BLK,N$FLG(R5) ; Enable I/O packet dequeuing

```

AXDSPP      CREATED BY MACRO ON 3-SEP-85 AT 11:00      PAGE 3      F 1

MACRO CROSS REFERENCE      CREF 04.00

| MACRO NAME | REFERENCES                                                     |
|------------|----------------------------------------------------------------|
| CALL       | 8-168      9-229      9-238      9-241      10-294      10-310 |
| CALLR      | 7-120      10-304      10-308                                  |
| CCBDF\$    | #6-75      6-78                                                |
| RESRG      | #6-75      10-311                                              |
| RETURN     | 7-119      8-183      9-246      10-313                        |
| SAVRG      | #6-75      10-309                                              |
| SLTDF\$    | #6-75      6-77                                                |
| SOB        | 10-289                                                         |

AXMDC - AUX MODEM CONTROLLER  
MODEM CONTROLLER TIMER SERVICE

MACRO V05.03b Friday 28-Jun-85 18:30<sup>F 2</sup> Page 13-1

|     |        |        |        |        |        |                     |                                       |
|-----|--------|--------|--------|--------|--------|---------------------|---------------------------------------|
| 370 | 000412 | 116501 | 000003 |        | MOVB   | M.STT(R5),R1        | ; GET CURRENT LINE STATE              |
| 371 | 000416 |        |        |        | CALL   | @MDMSRV(R1)         | ; DISPATCH TO SERVICING ROUTINE       |
| 372 | 000422 | 116565 | 000004 | 000005 | MOVB   | M.CSV(R5),M.PSV(R5) | ; UPDATE PREVIOUS SERVICE STATUS      |
| 373 | 000430 | 142765 | 000200 | 000004 | BICB   | #MC.CCB,M.CSV(R5)   | ; CLEAR CCB ALLOCATION FAILURE STATUS |
| 374 | 000436 | 062705 | 000010 |        | ADD    | #M.LEN,R5           | ; POINT TO NEXT ENTRY                 |
| 375 | 000442 | 000722 |        | 40\$:  | BR     | MDMSCN              | ; AND LOOP                            |
| 376 | 000444 |        |        | 50\$:  | RETURN |                     |                                       |
| 377 |        |        |        | :      |        |                     |                                       |

AXMDC - AUX MODEM CONTROLLER  
STUBC - WAIT FOR RING DETECT

MACRO V05.03b Friday 28-Jun-85 18:30<sup>G 2</sup> Page 14

## Symbol table

|                  |                  |                 |                 |                  |
|------------------|------------------|-----------------|-----------------|------------------|
| ASSCHK= 000000   | CS.DIS= 000040   | D\$ISK= 000000  | G\$TSS= 000000  | L.OWNR 000021    |
| ASSCPS= 000000   | CS.ENA= 000001   | D\$SL11= 000001 | G\$T7TK= 000000 | L.UNT 000013     |
| ASSPRI= 000000   | CS.ENB= 000020   | D\$SYNC= 000000 | G\$WRD= 000000  | MC.CAR= 000001   |
| ASSTRP= 000000   | CS.ERR= 100000   | D\$SYNM= 000000 | I\$RAR= 000000  | MC.CCB= 000200   |
| CB.CCB= 000002   | CS.FTL= 001000   | ENABLE 000114R  | I\$RDN= 000000  | MC.DSR= 000002   |
| CB.DDM= 000040   | CS.HCR= 000001   | E\$XPR= 000000  | K\$CNT= 177546  | MC.RNG= 000004   |
| CB.DLC= 000020   | CS.HFE= 002000   | FC.CCP= 000020  | K\$CSR= 177546  | MDMCTL 000022RG  |
| CB.RDB= 000004   | CS.LST= 040000   | FC.CTL= 000006  | K\$LDL= 000000  | MDMSCN 000310RG  |
| CB.SDB= 000010   | CS.MTL= 004000   | FC.KCP= 000016  | K\$TPS= 000074  | MDMSRV 000000R   |
| CB.SLI= 000100   | CS.RNG= 000010   | FC.KIL= 000004  | LD\$LP= 000000  | MS.END= 000200   |
| CB.XLB= 000001   | CS.ROV= 000004   | FC.MAN= 000024  | LF.ACT= 100000  | MS.SCA= 000002   |
| CCBGT = ***** GX | CS.RSN= 010000   | FC.MLD= 000026  | LF.BRO= 000400  | MS.SYN= 000001   |
| CC.LLC= 000200   | CS.SHU= 000001   | FC.PCT= 000030  | LF.BWT= 000007  | M\$CRB= 000124   |
| CE.ABO= 100362   | CS.SID= 000002   | FC.PWR= 000022  | LF.ENA= 002000  | M\$CRX= 000000   |
| CE.DAO= 100346   | CS.STR= 000004   | FC.RCE= 000002  | LF.LPB= 001000  | M\$FCS= 000000   |
| CE.DIS= 100366   | CS.SUC= 000001   | FC.RCP= 000014  | LF.MDC= 000100  | M\$MGE= 000000   |
| CE.ERR= 100370   | CS.TMO= 020000   | FC.TIM= 000010  | LF.MFL= 000400  | M\$NET= 000000   |
| CE.ILN= 100350   | CS.XUR= 000004   | FC.XCP= 000012  | LF.MTP= 000020  | M\$OVR= 000000   |
| CE.LTO= 100356   | CTLDSP 000060R   | FC.XME= 000000  | LF.PAC= 000200  | M.CSV 000004     |
| CE.MOP= 100372   | C\$CKP= 000000   | FS.AST= 000000  | LF.RDY= 040000  | M.LEN 000010     |
| CE.NTE= 100361   | C\$ORE= 000400   | FS.CIB= 002000  | LF.REA= 010000  | M.LLN 000001     |
| CE.RTE= 100376   | C\$RSH= 177564   | FS.CRA= 001000  | LF.SER= 000040  | M.PSV 000005     |
| CE.SRC= 100364   | C.ADD 000034     | FS.DIS= 013000  | LF.TIM= 000010  | M.STA 000006     |
| CE.STP= 100352   | C.BID 000003     | FS.DVC= 001000  | LF.UNL= 020000  | M.STS 000000     |
| CE.TME= 100354   | C.BUF 000014     | FS.ENB= 012000  | LF.X2P= 000000  | M.STT 000003     |
| CE.TMO= 100374   | C.BUF1 000014    | FS.EXI= 001000  | LN.CLO= 000000  | M.TIM 000002     |
| CE.UNS= 100344   | C.BUF2 000024    | FS.GET= 006000  | LN.DUM= 000005  | N\$ACC= 000001   |
| CF.CHN= 000001   | C.CNT 000020     | FS.HLT= 000000  | LN.LOA= 000004  | N\$BUF= 000001   |
| CF.EOM= 000004   | C.CNT1 000020    | FS.INJ= 000000  | LN.LOO= 000003  | N\$LDV= 000001   |
| CF.HDR= 000020   | C.CNT2 000030    | FS.KIL= 000000  | LN.OAU= 000003  | N\$MCP= 000001   |
| CF.LB= 100000    | C.FLG 000022     | FS.LCL= 100000  | LN.OFF= 000001  | N\$MLL= 000001   |
| CF.LIN= 000002   | C.FLG1 000022    | FS.LTM= 001000  | LN.ON= 000000   | N\$MOV= 000010   |
| CF.SOM= 000010   | C.FLG2 000032    | FS.MNT= 004000  | LN.OOP= 000004  | N\$NCT= 000001   |
| CF.SYN= 000040   | C.FNC 000010     | FS.MSN= 014000  | LN.OPE= 000001  | N\$PEM= 000001   |
| CF.TRN= 000100   | C.LIN 000006     | FS.REA= 001000  | LN.REF= 000002  | PDVTA = ***** GX |
| CM.CIR= 000002   | C.LNK 000000     | FS.RET= 000000  | LN.SER= 000002  | P\$P45= 000000   |
| CM.FMT= 100000   | C.MOD 000011     | FS.REZ= 003000  | LN.STA= 000017  | P\$WRD= 000000   |
| CM.HRD= 000002   | C.NSP 000004     | FS.RLB= 002000  | LN.SUB= 000360  | Q\$OPT= 000010   |
| CM.LIN= 000000   | C.PRO 000042     | FS.RNG= 011000  | LN.TRI= 000006  | RING 000066R     |
| CM.LOO= 000001   | C.RSV 000002     | FS.RST= 000000  | L\$ASG= 000000  | R\$DER= 000000   |
| CM.XLO= 000004   | C.STA 000007     | FS.RTN= 001000  | L\$DRV= 000000  | R\$K11= 000001   |
| CNCMP 000610R    | C.STR 000012     | FS.SET= 005000  | L\$P1R= 000001  | R\$SND= 000000   |
| CP.DCF= 000040   | C.URM 177776     | FS.SFC= 005000  | L\$11R= 000000  | R\$11M= 000000   |
| CP.HDL= 000007   | C.XACP 000004    | FS.SFR= 006000  | L.COST 000015   | SF.ACT= 000200   |
| CP.PS= 177400    | C.XID 000035     | FS.SFS= 004000  | L.CTL 000012    | SF.ENA= 000100   |
| CP.PSI= 000200   | C.XLEN 000044    | FS.SPW= 040000  | L.CVA 177776    | SF.LPB= 000004   |
| CP.XCF= 000100   | C.XPLI 000040    | FS.STM= 000000  | L.DDM 000002    | SF.MFL= 000040   |
| CP.2FR= 000030   | C.XPT 000034     | FS.STP= 002000  | L.DDS 000004    | SF.PAC= 000020   |
| CS.ABO= 000100   | C.XSVC 000042    | FS.STR= 001000  | L.DLC 000003    | SF.REA= 000010   |
| CS.BRO= 000002   | C.XTC 000037     | FS.TRM= 003000  | L.DLM 000006    | SF.SER= 000001   |
| CS.BUF= 000200   | C.X25 000036     | FS.WLB= 001000  | L.DLS 000010    | SF.SVC= 000002   |
| CS.CES= 000002   | DDAST = ***** GX | FS.XKL= 002000  | L.FLG 000000    | SF.UNL= 000040   |
| CS.CHN= 000010   | DDCCP = ***** GX | FS.XOF= 010000  | L.KRBA 000016   | SLTMA = ***** GX |
| CS.CMP= 000200   | DDMSN = ***** GX | FS.XON= 007000  | L.LEN= 000022   | ST.ABO 000020 G  |
| CS.DCR= 000400   | DISC 000756R     | FS.ZER= 002000  | L.MPF 000022    | ST.ACT 000010 G  |
| CS.DEF= 000004   | DSABLE 000134R   | F\$LVL= 000001  | L.NMST 000020   | ST.ADL 000012 G  |
| CS.DEV= 000002   | D\$BUG= 177514   | G\$TPP= 000000  | L.NSTA 000014   | ST.CDL 000006 G  |

```

185 000170 45$: CALL @STDLC ; SET UP DLC INDEX AND LINE TABLE ADDRESS
186 000174 CALL @PDSPDL ; DISPATCH THE PROCESS
187 000200 MTPS #0 ; ENABLE INTERRUPTS
188 000204 000700 BR 5$; GO EXAMINE QUEUE ONCE MORE
189
190 ;
191 ; GIVE UP PROCESSOR TO ALLOW OTHER FORK BLOCKS TO EXECUTE
192 ;
193
194 000206 016704 000000G 47$: MOV CMFRK,R4 ;::: POINT TO START OF FORK BLOCK
195 000212 005014 CLR (R4) ;::: CLEAR LINK WORD
196
197 .IF DF R$$MPL
198 CALL @QFORK ;::: QUEUE UP THE FORK BLOCK
199
200 .IFF
201
202 000214 010477 000002G MOV R4,@$FRKHD+2 ;::: PUT FORK BLOCK AT THE
203 000220 010457 000002G MOV R4,$FRKHD+2 ;::: END OF THE FORK LIST
204
205 .ENDC
206
207 000224 000401 BR 55$;::: EXIT FORK PROCESSING
208
209 ;
210 ; EXIT SOFTWARE LEVEL SERVICE
211 ;
212
213 000226 014343 50$: MOV -(R3),-(R3) ;::: FREE FORK BLOCK
214 000230 042777 100000 000000G 55$: BIC #CF,FRK,@CXOPT ;::: FORK PROCESSING NO LONGER ACTIVE
215
216 .IF DF R$$MPL
217 .IF NDF R$$PRO
218
219 BIT #F2.MP,@FMSK2 ;::: IS THIS A MULTIPROCESSOR?
220 BEQ 60$;::: BR IF NO
221 CALL @($P)+ ;::: CO-ROUTINE RETURN TO UNLOCK PROCESS QUEUE
222
223 60$:
224 .ENDC
225 .ENDC
226
227 000236 MTPS #0 ;::: ENABLE INTERRUPTS
228 000242 RETURN ; RETURN TO SYSTEM
229
230 .END
231 000001

```



```

229 000114 0i1046 20$: MOV (R0),-(SP) ; save next block address
230 000116 010346 MOV R3, -(SP) ; set up for $ceacc (prev blk)
231 000120 CALL @CEACC ; map address of block
232 000124 012600 MOV (SP)+,R0 ; get mapped address
233 000126 012610 MOV (SP)+,(R0) ; unlink block from list
234 000130 010200 MOV R2,R0 ; and return unmapped address to user
235 000132 RESMAP ; restore apr 6 mapping
236 000136 016777 177640 000000G MOV XAVLL,@XAVL ; update cex copy of listhead
237 000144 012603 MOV (SP)+,R3 ; restore register
238 000146 RETURN ;
239
240 .ENDC

```

\*\*FILE\*\*ID\*\*AXTIM

```

AAAAAA XX XX TTTTTTTTTT IIIII MM MM
AAAAAA XX XX TTTTTTTTTT IIIII MM MM
AA AA XX XX TT II MMMM MMMM
AA AA XX XX TT II MMMM MMMM
AA AA XX XX TT II MM MM MM
AA AA XX XX TT II MM MM MM
AA AA XX XX TT II MM MM MM
AAAAAAA XX XX TT II MM MM MM
AAAAAAA XX XX TT II MM MM MM
AA AA XX XX TT II MM MM
AA AA XX XX TT II MM MM
AA AA XX XX TT IIIII MM MM
AA AA XX XX TT IIIII MM MM

```

```

LL SSSSSSS TTTTTTTTTT
LL SSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSS TT
LL SSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLL SSSSSSS TT
LLLLLLLLL SSSSSSS TT

```

```

419 .SBTTL PUT TIMER ENTRY ON QUEUE
420 :+
421 : **--STALT-PUT TIMER ENTRY ON QUEUE
422 :
423 : LINK A SHORT TIMER ENTRY ONTO A 100 MSEC TIMER QUEUE.
424 :
425 : INPUTS:
426 : R0 - POINTER TO TIMER QUEUE LISTHEAD
427 : R4 - POINTER TO SHORT TIMER ENTRY
428 :
429 STALT: CLR (R4) ; CLEAR LINK WORD
430 INHIB$;;; INHIBIT INTERRUPTS
431
432 .IF DF R$$MPL
433 .IF NDF R$$PRO
434
435 BIT #F2.MP,@FMSK2 ;;; IS THIS A MULTI-PROCESSOR?
436 BEQ 10$;;; BR IF NO
437 CALL @MPLCK ;;; LOCK ACCESS TO COMMESEC RESOURCES
438
439 10$: .ENDC
440 .ENDC
441
442 MOV R4,@2(R0) ;;; LINK ENTRY TO END OF LIST
443 MOV R4,2(R0) ;;; ...
444
445 .IF DF R$$MPL
446 .IF NDF R$$PRO
447
448 BIT #F2.MP,@FMSK2 ;;; IS THIS A MULTI-PROCESSOR?
449 BEQ 20$;;; BR IF NO
450 CALL @(SP)+ ;;; CO-ROUTINE RETURN TO UNLOCK RESOURCES
451
452 20$: .ENDC
453 .ENDC
454
455 ENABL$; ENABLE INTERRUPTS
456 RETURN

```

```

113 .SBTTL Connect QIO processing
114
115 +
116 ***-CON-Connect request processing
117
118 This routine processes outgoing connect requests.
119
120 Inputs:
121 R1 = Address of the task's header
122 R2 = I/O subfunction code/4
123 R3 = Address of I/O packet
124 R4 = Address of the window block
125 R5 = Address of database descriptor
126
127 Outputs:
128 'C' Clear - Connect successfully processed
129 'C' Set - Connect failure
130
131 000106 010364 000010 CON: MOV R3,W.TMP(R4) ; Save address of I/O packet
132
133 000112 MAP I.PRM(R3) ; Map to the connect block
134 000120 016302 000002G MOV I.PRM+2(R3),R2 ; Get virtual address of buffer
135 000124 012701 000000G MOV #CNBLK,R1 ; Set address of internal connect block
136 000130 012700 000032 MOV #N.RIDC-N.RND,R0 ; Set up # of bytes to copy
137 000134 112221 10$: MOV (R2)+(R1)+ ; Copy first part of connect block
138 000136 SOB R0,10$; ...
139
140 000142 012701 000000G MOV #REQID,R1 ; Set up new address of internal connect block
141 000146 012700 000056 MOV #N.RQL-N.RIDC,R0 ; Set up # of bytes to copy
142 000152 112221 20$: MOV (R2)+(R1)+ ; Copy second part of connect block
143 000154 SOB R0,20$; ...
144
145 000160 RECMAP ; Recover high APR mapping
146 000166 112767 000001 000000G MOV #NC.FM1,$SRDFM ; Source format is always 1
147 000174 105067 000000G CLR $SROBJ ; Source object is always 0
148 000200 012702 000000G MOV #SRDSC,R2 ; Point to source process/task descriptor
149 000204 012722 000006 MOV #6,(R2)+ ; 6 bytes long
150 000210 016301 000000G MOV I.TCB(R3),R1 ; Get task name
151 000214 016146 000002G MOV T.NAM+2(R1),-(SP)
152 000220 016101 000000G MOV T.NAM(R1),R1 ; ...
153 000224 CALL C5TA ; Convert to ascii
154 000230 012601 MOV (SP)+,R1 ; ...
155 000232 CALL C5TA ; ...
156
157 000236 004167 000000G JSR R1,CPYOPT ; Copy optional user data
158 000242 000004G .WORD I.PRM+4
159
160 000244 CALL SNDCON ; Transmit the connect request
161 000250 103405 BCS 30$; If CS, failure
162 000252 016700 000000G MOV $MAIBX,R0 ; Recover address of mailbox
163 000256 105260 000010 INCB M.USE(R0) ; Update count of active/pending logical links
164 000262 RETURN
165
166 000264 012700 000000C 30$: MOV #IE.NRJ&377,R0 ; Set network reject code
167 000270 015501 000022 MOV #SERRC(R5),R1 ; Get the reject reason code
168 000274 CALL IODUN ; Complete the request in error

```

\*\*FILE\*\*ID\*\*SECTL

```

SSSSSSSS EEEEEEEEE SSSSSSSS CCCCCCCC TTTTTTTTTT LL
SSSSSSSS EEEEEEEEE SSSSSSSS CCCCCCCC TTTTTTTTTT LL
SS EE SS CC TT LL
SS EE SS CC TT LL
SS EE SS CC TT LL
SS EE SS CC TT LL
SSSSSSS EEEEEEEEE SSSSSSSS CC TT LL
SSSSSSS EEEEEEEEE SSSSSSSS CC TT LL
 SS EE SS CC TT LL
 SS EE SS CC TT LL
 SS EE SS CC TT LL
 SS EE SS CC TT LL
SSSSSSSS EEEEEEEEE SSSSSSSS CCCCCCCC TT LLLLLLLLLL
SSSSSSSS EEEEEEEEE SSSSSSSS CCCCCCCC TT LLLLLLLLLL

```

```

11 11 SSSSSSSS
11 11 SSSSSSSS
1111 1111 SS
1111 1111 SS
11 11 SS
11 11 SS
11 11 SSSSSS
11 11 SSSSSS
11 11 SS
11 11 SS
11 11 SS
11 11 SS
111111 111111 SSSSSSSS
111111 111111 SSSSSSSS

```

```

394 .SBTTL Find requested mail item
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416 000244 .PSECT $HIGH
417
418 000244 062704 000014 FNDMAI: ADD #M.MAIL,R4 ; Point to mailbox listhead
419
420 000250 011401 10$: MOV (R4),R1 ; Get next item on list
421 000252 001432 BEQ 50$; If EQ, no more
422 000254 116100 000010 MOVB C.FNC(R1),R0 ; Get the function code
423 000260 100006 BPL 20$; If PL, not a connect type request
424 000262 132761 000004 000011 BITB #CX.UNL,C.MOD(R1) ; Has this request been de-queued already?
425 000270 001025 BNE 60$; If NE, yes
426
427 000272 042700 177600 BIC #^C<177>,R0 ; Isolate function code
428
429 000276 120227 000006 20$: CMPB R2,#6 ; Is this a normal GND$?
430 000302 001415 BEQ 40$; If EQ, yes
431
432 000304 105763 000006G TSTB I.PRM+6(R3) ; Type mask specified?
433 000310 001403 BEQ 30$; If EQ, no
434 000312 126300 000006G CMPB I.PRM+6(R3),R0 ; Match on type code
435 000316 001012 BNE 60$; If NE, no
436
437 000320 105763 000007G 30$: TSTB I.PRM+7(R3) ; LUN mask specified?
438 000324 001404 BEQ 40$; If EQ, no
439 000326 126361 000007G 000011 CMPB I.PRM+7(R3),C.MOD(R1) ; If NE, no LUN match
440 000334 001003 BNE 60$
441
442 000336 005727 40$: TST (PC)+ ; Indicate success
443 000340 000261 50$: SEC ; Indicate failure
444 000342 RETURN
445
446 000344 010104 60$: MOV R1,R4 ; Move on down the list
447 000346 000740 BR 10$; and try again

```

F 11

SE\$CTL - Session control contro MACRO V05.03b Friday 28-Jun-85 19:53 Page 20-3  
 Symbol table

|                  |                |                |                 |                   |     |
|------------------|----------------|----------------|-----------------|-------------------|-----|
| T\$KMG= 000000   | VE.FAI= 177777 | WK.RCV= 000004 | W.LLT 000004    | \$CALLX= ***** GX |     |
| T\$MIN= 000000   | VFY 000446R    | WK.SND= 000002 | W.LUN 000003    | \$CLQIO 000000RG  |     |
| T.NAM = ***** GX | VS.NPV= 000001 | WS.DIP= 000010 | W.MBOX 000012   | \$CTQIO 000000RG  | 002 |
| UISAR6= ***** GX | VS.PRV= 000002 | WS.INT= 000002 | W.RCVQ 000024   | \$IOPKT= ***** GX |     |
| US\$CNF= 000002  | VZ.NVD= 000000 | WS.KAS= 000004 | W.SEGZ 000006   | \$MAIBX= ***** GX |     |
| US\$DIS= 000006  | V\$CTR= 001000 | WS.STA= 000001 | W.SNDQ 000016   | \$RQNAM= ***** GX |     |
| US\$OON= 000000  | WK.ACK= 000001 | W.CINT 000022  | W.STAT 000002   | \$UCB = ***** GX  |     |
| US\$DSC= 000004  | WK.AST= 000200 | W.CSND 000020  | W.TMP 000010    | \$SSHFT= 000001   |     |
| US\$EAC= 000012  | WK.DIS= 000010 | W.CTL 000000   | W.WBL 000026    | \$\$\$ = 000062   |     |
| US\$WOS= 000010  | WK.INT= 000020 | W.KAST 000014  | X\$SDBT= 000000 | .\$\$\$\$= 000034 |     |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
 001364 001 (RW,I,LCL,REL,CON)  
 \$HIGH 000504 002 (RW,I,LCL,REL,CON)  
 Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 100  
 Work file writes: 102  
 Size of work file: 25667 Words ( 101 Pages)  
 Size of core pool: 17608 Words ( 67 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:35.46  
 SY:SE\$CTL11S.V2,[131,134]SE\$CTL11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCS/PA:1,[131,10]V2,SE\$CTL

SESCTR - Session control counts MACRO V05.03b Friday 28-Jun-85 19:54 Page 9-2  
Symbol table

|                 |                  |                 |                 |                  |
|-----------------|------------------|-----------------|-----------------|------------------|
| FC.PWR= 000022  | G\$STSS= 000000  | L.LLA 000002    | MO\$25P= 000002 | N\$SSES= 000001  |
| FC.RCE= 000002  | G\$STTK= 000000  | L.LNG 000124    | MO\$25S= 000004 | OP\$INI= 000000  |
| FC.RCP= 000014  | G\$SWRD= 000000  | L.LNO 000026    | MO\$29S= 000010 | OP\$TER= 000001  |
| FC.TIM= 000010  | IN.DAT= 000400   | L.LPT 000065    | MSHIGH= 000003  | PH\$HDE= 000004  |
| FC.XCP= 000012  | IN.ILS= 000001   | L.LSA 000030    | MS\$CRB= 000124 | PH\$LOC= 000002  |
| FC.XMC= 000000  | IS\$RAR= 000000  | L.LSFD 000046   | MS\$CRX= 000000 | PH\$MTS= 000003  |
| FR\$BCC= 000007 | IS\$RDN= 000000  | L.LSFI 000044   | MS\$FCS= 000000 | PH\$MUP= 000000  |
| FR\$CCF= 000001 | KISAR6= ***** GX | L.LTT 000062    | MS\$MGE= 000000 | PH\$WCS= 000001  |
| FR\$CDF= 000002 | K\$SCNT= 177546  | L.MASQ 000070   | MS\$MUP= 000000 | P\$P45= 000000   |
| FR\$DAO= 000011 | K\$SCSR= 177546  | L.MAST 000073   | MS\$NET= 000000 | P\$SWRD= 000000  |
| FR\$EXC= 000000 | K\$SLDC= 000000  | L.MASZ 000072   | MS\$OVR= 000000 | Q\$SOPT= 000010  |
| FR\$FRM= 000010 | K\$STPS= 000074  | L.NIN 000020    | MS\$100= 000000 | RE\$ADC= 000004  |
| FR\$FTL= 000005 | LA.ACK= 100000   | L.NXN 000016    | MS\$101= 000001 | RE\$ADF= 000017  |
| FR\$OPN= 000004 | LA.CRS= 020000   | L.NXTH 000010   | MS\$102= 000002 | RE\$ADR= 000007  |
| FR\$RFD= 000006 | LA.MSK= 170000   | L.OPD 000103    | MS\$103= 000003 | RE\$BLK= 000010  |
| FR\$SBU= 000012 | LA.NAK= 110000   | L.OPDL 000102   | NC.FM0= 000000  | RE\$CAF= 000014  |
| FR\$SHD= 000003 | LA.NMS= 010000   | L.REM 000006    | NC.FM1= 000001  | RE\$DAT= 000001  |
| FR\$SBU= 000013 | LA.RES= 040000   | L.RFC 000050    | NC.FM2= 000002  | RE\$DRP= 000016  |
| FR\$UPT= 000014 | LA.WND= 004000   | L.RLA 000004    | NF\$BLK= 000100 | RE\$LDLT= 000013 |
| FS.AST= 000000  | LD\$LP= 000000   | L.RND 000022    | NF\$DMO= 000010 | RE\$LSN= 000012  |
| FS.CIB= 002000  | LF.DRD= 000004   | L.RTQ 000060    | NF\$MOU= 000040 | RE\$NML= 000001  |
| FS.CRA= 001000  | LF.FRC= 000001   | L.RTYD 000055   | NF\$RST= 000002 | RE\$OPR= 000004  |
| FS.DIS= 013000  | LF.HFO= 000010   | L.RTYI 000057   | NF\$SCN= 000020 | RE\$OPR= 000000  |
| FS.DVC= 001000  | LF.HMF= 000040   | L.SEC 000064    | NF\$SHU= 000004 | RE\$RCV= 000001  |
| FS.ENB= 012000  | LF.HSF= 000020   | L.SEGZ 000076   | NF\$TIM= 000200 | RE\$SED= 000011  |
| FS.EXI= 001000  | LF.IRD= 000002   | L.STA 000000    | NS\$DON= 000000 | RE\$SKW= 000006  |
| FS.GET= 006000  | LF.MMF= 000200   | L.TC 000042     | NS\$SDI= 000002 | RE\$STA= 000002  |
| FS.HLT= 000000  | LF.MSF= 000100   | L.TIC 000043    | NS\$WDC= 000004 | RE\$SUM= 000003  |
| FS.INI= 000000  | LS.DLS= 100000   | L.TIPD 000013   | NS\$ACQ= 000000 | RE\$SYN= 000000  |
| FS.KIL= 000000  | LS.FCC= 000004   | L.TIPI 000012   | NS\$ACTL 000032 | RE\$TME= 000021  |
| FS.LCL= 100000  | LS.FCO= 000001   | L.TMRD 000054   | NS\$CIR 000034  | RE\$TMD= 000000  |
| FS.LTM= 001000  | LS.FCI= 000002   | L.TMRI 000056   | NS\$DLA 000020  | RE\$TMR= 000020  |
| FS.MNT= 004000  | LS.ILS= 100000   | L.TYP 000001    | NS\$DLV 000014  | RE\$UPT= 000002  |
| FS.MSN= 014000  | LS.MAK= 000020   | L.USA 000024    | NS\$LEN 000054  | RE\$URE= 000003  |
| FS.REA= 001000  | LS.MNK= 000040   | L.USTA 000036   | NS\$ENC 000042  | RE\$VER= 000005  |
| FS.RET= 000000  | LS.RES= 000360   | L.VER 000015    | NS\$ERRC 000022 | RE\$VRQ= 000015  |
| FS.REZ= 003000  | LS.RSV= 000300   | L.WIND 000040   | NS\$FLG 000005  | RT\$INI= 000002  |
| FS.RLB= 002000  | LT.CCA= 000020   | MA.CI= 000040   | NS\$FNC 000006  | RT\$OFF= 000001  |
| FS.RNG= 011000  | LT.DIR= 000010   | MA.DA= 000000   | NS\$GENQ 000052 | RT\$ON= 000000   |
| FS.RST= 000000  | LT.LCL= 000001   | MA.IL= 000020   | NS\$GTM 000015  | RS\$DER= 000000  |
| FS.RTN= 001000  | LT.LPL= 000002   | MC.CC= 000040   | NS\$HIGH 000033 | RS\$K11= 000001  |
| FS.SET= 005000  | LT.NOT= 000040   | MC.CI= 000020   | NS\$LLT 000026  | RS\$SND= 000000  |
| FS.SFC= 005000  | LT.RSU= 000200   | MC.DC= 000100   | NS\$LLTM 000024 | RS\$11M= 000000  |
| FS.SFR= 006000  | LT.SLI= 000004   | MC.DI= 000060   | NS\$LVC 000036  | RS\$11S= 000000  |
| FS.SFS= 004000  | LT.TDA= 000100   | MC.NO= 000000   | NS\$MBXQ 000050 | SC\$OFF= 000001  |
| FS.SPW= 004000  | LS\$ASG= 000000  | MC.RC= 000140   | NS\$PLLT 000030 | SC\$ON= 000000   |
| FS.STM= 000030  | LS\$DRV= 000000  | MD.BM= 000040   | NS\$SLA 000016  | SC\$RST= 000003  |
| FS.STP= 002000  | LS\$P11= 000001  | MD.EM= 000100   | NS\$SND 000012  | SC\$SHU= 000002  |
| FS.STR= 001000  | LS\$11R= 000000  | MD.ILS= 000040  | NS\$TIM 000004  | ST\$CC= 000004   |
| FS.TRM= 003000  | L.CSTA 000037    | MD.IM= 000020   | NS\$VCB 000010  | ST\$CIR= 000006  |
| FS.WLB= 001000  | L.CTR 000074     | MF.ACK= 000004  | NS\$VCC= 000001 | ST\$CIS= 000002  |
| FS.XKL= 002000  | L.DCR 000100     | MF.CTL= 000010  | NS\$EVL= 000001 | ST\$DAT= 000010  |
| FS.XOF= 010000  | L.FLAG 000014    | MF.DAT= 000000  | NS\$LDV= 000001 | ST\$DIP= 000012  |
| FS.XON= 007000  | L.LISQ 000052    | MO\$SAC= 000016 | NS\$MLL= 000001 | ST\$PND= 000014  |
| FS.ZER= 002000  | L.LITT 000066    | MO\$SPR= 000012 | NS\$MOV= 000010 | SV\$DUM= 000001  |
| F\$SLVL= 000001 | L.LDA 000032     | MO\$SSV= 000014 | NS\$NCT= 000001 | SV\$LOA= 000000  |
| G\$STPP= 000000 | L.LIA 000034     | MO\$25A= 000006 | NS\$PEM= 000001 | S\$SWRG= 000000  |



F 13

SESDAT - Session control local MACRO V05.03b Friday 28-Jun-85 19:54 Page 7-1

Local data

|     |        |     |     |     |                          |                                      |
|-----|--------|-----|-----|-----|--------------------------|--------------------------------------|
| 105 | 000054 | 044 | 110 | 117 | HOST:: .ASCII /\$HOST /  | ; Reserved node name to host system  |
|     | 000057 | 123 | 124 | 040 |                          |                                      |
| 106 | 000062 | 044 | 114 | 117 | LOCAL:: .ASCII /\$LOCAL/ | ; Reserved node name to local system |
|     | 000065 | 103 | 101 | 114 |                          |                                      |

G 13

SESDAT - Session control local MACRO V05.03b Friday 28-Jun-85 19:54 Page 8

```

83 .SBTTL Disconnect/abort QIO processing
84 +
85 **--DSC-Disconnect QIO processing
86 **--ABT-Abort QIO processing
87
88 This routine processes all requests for disconnecting a logical link.
89 the disconnect request will be delayed until all pending transmits
90 have completed.
91
92 Inputs:
93 R1 = Address of task's header
94 R3 = Address of I/O packet
95 R4 = Address of window block
96 R5 = Address of database descriptor
97
98 .PSECT
99
100 .ENABL LSB
101
102 000000 005067 000000G DSC: CLR $REASN ; Set reason code to synchronous disconnect by user
103 000004 005764 000016 TST W.SNDQ(R4) ; Any transmits pending?
104 000010 001405 BEQ 10$; If EQ, no ... continue processing
105 000012 CALLR IOREDO ; Re-queue I/O packet for later processing
106
107 000016 012767 000011 000000G ABT: MOV #ER$ABT,$REASN ; Set reason code to user abort
108
109 000024 010364 000010 10$: MOV R3,W.TMP(R4) ; Save address of disconnect I/O packet
110 000030 CALL FLSHIO ; Flush away pending I/O requests
111 000034 116400 000003 MOVB W.LUN(R4),R0 ; Get LUN for this link
112 000040 016403 000012 MOV W.MBOX(R4),R3 ; Get address of mailbox
113 000044 CALL FLSHMB ; Flush events from mailbox
114
115 000050 004167 000000G JSR R1,CPYOPT ; Copy optional data to internal buffer
116 000054 000000G .WORD I.PRM
117
118 000056 016403 000004 MOV W.LLT(R4),R3 ; Get physical address of LLT
119 000062 CALLE ACCLLT ; Gain access to the LLT
120
121 000072 CALLR SNDDIS ; Disconnect the link
122
123 .DSABL LSB

```

|                  |                 |                  |                 |                   |
|------------------|-----------------|------------------|-----------------|-------------------|
| MO\$SAC= 000016  | NS\$SDI= 000002 | N\$LLTM 000024   | RE\$ADR= 000007 | SC\$ON = 000000   |
| MO\$SPR= 000012  | NS\$WDC= 000004 | NSLVC 000036     | RE\$BLK= 000010 | SC\$RST= 000003   |
| MO\$SSV= 000014  | NT\$AKD= 000020 | NSMBXQ 000050    | RE\$CAF= 000014 | SC\$SHU= 000002   |
| MO\$25A= 000006  | NT\$AKI= 000022 | NSPLLT 000030    | RE\$DAT= 000001 | ST\$CC = 000004   |
| MO\$25P= 000002  | NT\$CC = 000016 | N\$SLA 000016    | RE\$DRP= 000016 | ST\$CIR= 000006   |
| MO\$25S= 000004  | NT\$CON= 000000 | NS\$NOD 000012   | RE\$LDT= 000013 | ST\$CIS= 000002   |
| MO\$29S= 000010  | NT\$CTL= 000000 | NS\$TIM 000004   | RE\$LSN= 000012 | ST\$DAT= 000010   |
| MS\$HIGH= 000003 | NT\$DAT= 000002 | NSVCB 000010     | RE\$NML= 000001 | ST\$DIP= 000012   |
| MS\$CRB= 000124  | NT\$DC = 000012 | NS\$ACC= 000001  | RE\$OPE= 000004 | ST\$PND= 000014   |
| MS\$CRX= 000000  | NT\$DIS= 000014 | NS\$ACK= 000011  | RE\$OPR= 000000 | SV\$DUM= 000001   |
| MS\$FCS= 000000  | NT\$DLS= 000006 | NS\$EVL= 000001  | RE\$RCV= 000001 | SV\$LOA= 000000   |
| MS\$MGE= 000000  | NT\$ILS= 000010 | NS\$HDR= 000007  | RE\$SED= 000011 | SS\$WRG= 000000   |
| MS\$MUP= 000000  | NT\$IMS= 000002 | NS\$LDV= 000001  | RE\$SKW= 000006 | SS\$YSZ= 007600   |
| MS\$NET= 000000  | NT\$INT= 000004 | NS\$MLL= 000001  | RE\$STA= 000002 | TS\$KMG= 000000   |
| MS\$OVR= 000000  | NT\$RET= 000032 | NS\$MOV= 000010  | RE\$SUM= 000003 | TS\$MIN= 000000   |
| MS\$100= 000000  | NT\$ROU= 000024 | NS\$NCT= 000001  | RE\$SYN= 000000 | US\$CNF= 000002   |
| MS\$101= 000001  | NT\$RTR= 000030 | NS\$OVR= 000022  | RE\$TME= 000021 | US\$DIS= 000006   |
| MS\$102= 000002  | NT\$TSP= 000026 | NS\$PEM= 000001  | RE\$TMO= 000000 | US\$DON= 000000   |
| MS\$103= 000003  | NS\$ACQ 000000  | NS\$SES= 000001  | RE\$TMR= 000020 | US\$DSC= 000004   |
| NC.FMO= 000000   | NS\$ACTL 000032 | OP\$INI= 000000  | RE\$UPT= 000002 | US\$EAC= 000012   |
| NC.FM1= 000001   | NS\$CIR 000034  | OP\$TER= 000001  | RE\$URE= 000003 | US\$WDS= 000010   |
| NC.FM2= 000002   | NS\$DLA 000020  | PH\$HDE= 000004  | RE\$VER= 000005 | US.MDM= ***** GX  |
| NF\$BLK= 000100  | NS\$DLY 000014  | PH\$LOC= 000002  | RE\$VRQ= 000015 | U.STS = ***** GX  |
| NF\$DMO= 000010  | NS\$LEN 000054  | PH\$MTS= 000003  | RT\$INI= 000002 | V\$SCTR= 001000   |
| NF\$MOU= 000040  | NS\$ENC 000042  | PH\$UMP= 000000  | RT\$OFF= 000001 | WORD1 = 000200    |
| NF\$RST= 000002  | NS\$ERRC 000022 | PH\$WCS= 000001  | RT\$ON = 000000 | WORD2 = 000000    |
| NF\$SCN= 000020  | NS\$FLG 000005  | PS\$P4S= 000000  | R\$DER= 000000  | X\$SDBT= 000000   |
| NF\$SHU= 000004  | NS\$FNC 000006  | PS\$WRD= 000000  | R\$K11= 000001  | \$CALLX= ***** GX |
| NF\$TIM= 000200  | NS\$GENQ 000052 | QS\$OPT= 000010  | R\$SND= 000000  | \$DMQIO 000000RG  |
| NM\$ARA= 176000  | NS\$GTM 000015  | REJECT= ***** GX | R\$11M= 000000  | \$UCB = ***** GX  |
| NM\$NOD= 001777  | NS\$HIGH 000033 | RE\$ADC= 000004  | R\$11S= 000000  | \$SSHFT= 000001   |
| NS\$DON= 000000  | N\$LLT 000026   | RE\$ADF= 000017  | SC\$OFF= 000001 |                   |

. ABS. 000216 000 (RW,I,GBL,ABS,OVR)  
 000076 001 (RW,I,LCL,REL,CON)  
 \$HIGH 000126 002 (RW,I,LCL,REL,CON)  
 Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 86  
 Work file writes: 70  
 Size of work file: 20627 Words ( 81 Pages)  
 Size of core pool: 17608 Words ( 67 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:19.45

SY:SESDMO11S.V2,[131,134]SESDMO11S/CR/-SP=SY:[1,1]RSXMC.M.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMC/PA:1,[131,10]V2,SESDMO

```

155
156 000134 017700 000000G 60$: MOV @TKTCB,R0 ;; Point to receive listhead
157 000140 062700 000000G ADD #1.RCVL,R0 ;; ...
158 000144 @QRMVF ;; Dequeue an I/O packet
159 000150 103447 BCS 90$;; If CS, none present
160
161 000152 152765 000200 000005 BISB #NFSTIM,NFLG(R5) ;; Show we still need timer support
162 000160 010103 MOV R1,R3 ;; Copy I/O packet address
163 000162 010367 000000G MOV R3,$IOPKT ;; Save I/O packet address for completions
164
165 000166 016302 000000G MOV I.TCB(R3),R2 ;; Get tasks TCB address
166 000172 016202 000000G MOV T.PCB(R2),R2 ;; then PCB address for window 0
167 000176 016201 000000G MOV P.HDR(R2),R1 ;; Get address of task header
168
169 .IF DF X$$HDR
170
171 BNE 70$;; If NE, header is in primary pool
172 MAP P.REL(R2) ;; Map to the task header
173 MOV #140000,R1 ;; Set up virtual address of task header
174 70$: MOV @KSAR6,$HDRMP ;; Save header mapping
175
176 .ENDC
177
178 000202 116302 000000G MOVB I.FCN(R3),R2 ;; Get I/O subfunction code
179 000206 006202 ASR R2 ;; Form word index
180 000210 006202 ASR R2 ;; for dispatch
181 000212 042702 177761 BIC #^C<16>,R2 ;; Remove unwanted bits
182 000216 116300 000001G MOVB I.FCN+1(R3),R0 ;; Get the I/O function code
183 000222 162700 000010 SJB #10,R0 ;; Check for codes 6 and 7 (DMO, CLN)
184 000226 002402 BLT 80$;; If LT, ok
185
186 000230 162700 000023 SUB #23,R0 ;; Convert to ACP function code set
187
188 000234 006300 80$: ASL R0 ;; Form word index
189 000236 017304 000000G MOV @I.LN2(R3),R4 ;; Get address of window block
190 000242 042704 000001 BIC #1,R4 ;; or mailbox
191 000246 RECMAP ;; Recover high APR mapping
192 000254 CALL @Q1ODSP(R0) ;; Dispatch to processing routine
193
194 000260 132765 000100 000005 BITB #NFSBLK,NFLG(R5) ;; Is I/O packet dequeuing blocked?
195 000266 001671 BEQ 30$;; If EQ, no ... try for next CCB
196
197 000270 132765 000020 000005 90$: BITB #NFS$SCN,N$FLG(R5)
198 000276 001260 BNE 20$;; If NE, force extra scan of general delivery queue
199
200 000300 RECMAP ;; Recover high APR mapping
201 000306 CALL AUXTSK ;; Check if there any general task requests
202 ;; to be serviced
203
204 .IF DF N$$EXT
205
206 MOV N$MBXQ(R5),R0 ;; If there are any mailboxes
207 BIS N$GENQ(R5),R0 ;; general delivery queue entries
208 BISB N$ACTL(R5),R0 ;; or active logical links,
209 BNE 95$;; Don't exit yet
210 BITB #NFSDMO,NFLG(R5)
211 BNE 95$;; Normal processing if network marked for dismount

```



```

379 .SBTTL ST.WRG - WAIT FOR RING DETECT
380 +
381 ***.WRNG-WAIT FOR RING DETECT
382 :
383 :
384 :
385 :
386 :
387 :
388 :
389 :
390 000446 132765 000004 000004 .WRNG: BITB #MC.RNG,M.CSV(R5) ; HAS RING BEEN DETECTED?
391 000454 001410 BEQ 20$; NO ... CONTINUE WAITING
392 000456 012703 000010 10$: MOV #CS.RNG,R3 ; SET UP COMPLETION STATUS
393 000462 112702 000000 MOVB #ST.IDL,R2 ; NEW NEXT STATE
394 000466 012701 011000 MOV #FS.RNG,R1 ; SUB-FUNCTION TO BE USED
395 000472 CALLR $SCHPR ; AND SCHEDULE DLC PROCESS
396 000476 20$: RETURN
397 ;

```

|                   |                |                |                  |                   |
|-------------------|----------------|----------------|------------------|-------------------|
| ST.CER 000014 G   | TM.CON= 000002 | ZF.KMX= 000020 | ZS.BSY= 140000   | \$PABD 000216RG   |
| ST.DDL 000016 G   | TM.DIS= 000002 | ZF.LLC= 000004 | Z.AVL 000014     | \$SCHPR 001100RG  |
| ST.IDL 000000 G   | T\$KMG= 000000 | ZF.LMC= 000100 | Z.DAT 000016     | \$SCHP2 001106RG  |
| ST.WCN 000004 G   | T\$MIN= 000000 | ZF.MAN= 020000 | Z.DSP 000000     | .ACTD 000720R     |
| ST.WRG 000032 G   | V\$CTR= 001000 | ZF.MFL= 000010 | Z.FLG 000010     | .ACTV 000642R     |
| \$\$\$WRG= 000000 | X\$DBT= 000000 | ZF.MTM= 000400 | Z.LEN = 000016   | .COND 000554R     |
| \$\$\$YSZ= 007600 | ZF.COU= 001000 | ZF.MUX= 000040 | Z.LLN 000006     | .CONE 001006R     |
| S.COST 000001     | ZF.DDM= 000001 | ZF.PSE= 002000 | Z.MAP 000020     | .DISD 001040R     |
| S.FLG 000000      | ZF.DIA= 004000 | ZF.SLI= 010000 | Z.NAM 000004     | .PABD 001022R     |
| S.LEN 000004      | ZF.DLC= 000002 | ZF.TIM= 000200 | Z.PCB 000012     | .WCN 000500R      |
| S.NMST 000002     | ZF.DVP= 100000 | ZF.X3P= 000000 | Z.SCH 000007     | .WRNG 000446R     |
| S.OWNR 000003     | ZF.INI= 040000 | ZS.ASN= 100000 | \$MDCIN 000244RG | .\$\$\$\$= 000034 |
| TM.ACT= 000002    |                |                |                  |                   |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
 001142 001 (RW,I,LCL,REL,CON)

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 35  
 Work file writes: 39  
 Size of work file: 17700 Words ( 70 Pages)  
 Size of core pool: 17608 Words ( 67 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:23.38

SY:AXMDC.V2,[131,134]AXMDC/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]AXMDC

A\$\$CHK= 000000  
 A\$\$CPS= 000000  
 A\$\$PRI= 000000  
 A\$\$TRP= 000000  
 CB.CCR= 000002  
 CB.DDM= 000040  
 CB.DLC= 000020  
 CB.RDB= 000004  
 CB.SDB= 000010  
 CB.SLI= 000100  
 CB.XLB= 000001  
 CC.LLC= 000200  
 CE.ABO= 100362  
 CE.DAO= 100346  
 CE.DIS= 100366  
 CE.ERR= 100370  
 CE.ILN= 100350  
 CE.LTO= 100356  
 CE.MPD= 100372  
 CE.NTE= 100361  
 CE.RTE= 100376  
 CE.SRC= 100364  
 CE.STP= 100352  
 CE.TME= 100354  
 CE.TMO= 100374  
 CE.UNS= 100344  
 CF.CHN= 000001  
 CF.DDM= 000002  
 CF.DYN= 000004  
 CF.EIS= 000010  
 CF.EOM= 000004  
 CF.FRK= 100000  
 CF.HDR= 000020  
 CF.LB = 100000  
 CF.LIN= 000002  
 CF.LOG= 000020  
 CF.MDM= 000001  
 CF.SDM= 000010  
 CF.SYN= 000040  
 CF.TIM= 000400  
 CF.TRN= 000100  
 CM.FRK = \*\*\*\*\* GX  
 CM.CIR= 000002  
 CM.FMT= 100000  
 CM.HRD= 000002  
 CM.LIN= 000000  
 CM.LOO= 000001  
 CM.XLO= 000004  
 CP.DCF= 000040  
 CP.HDL= 000007  
 CP.PS = 177400

CP.PSI= 000200  
 CP.XCF= 000100  
 CP.2FR= 000030  
 CS.ABO= 000100  
 CS.BRO= 000002  
 CS.BUF= 000200  
 CS.CES= 000002  
 CS.CHN= 000010  
 CS.CMP= 000200  
 CS.DCR= 000400  
 CS.DEF= 000004  
 CS.DEV= 000002  
 CS.DIS= 000040  
 CS.ENA= 000001  
 CS.ENB= 000020  
 CS.ERR= 100000  
 CS.FTL= 001000  
 CS.HCR= 000001  
 CS.HFE= 002000  
 CS.LST= 040000  
 CS.MTL= 004000  
 CS.RNG= 000010  
 CS.ROV= 000004  
 CS.RSN= 010000  
 CS.SHU= 000001  
 CS.SID= 000002  
 CS.STR= 000004  
 CS.SUC= 000001  
 CS.TMO= 020000  
 CS.XUR= 000004  
 CXOPT = \*\*\*\*\* GX  
 C\$\$CKP= 000000  
 C\$\$ORE= 000400  
 C\$\$RSH= 177564  
 C.ADD= 000034  
 C.BID= 000003  
 C.BUF= 000014  
 C.BUF1= 000014  
 C.BUF2= 000024  
 C.CNT= 000020  
 C.CNT1= 000020  
 C.CNT2= 000030  
 C.FLG= 000022  
 C.FLG1= 000022  
 C.FLG2= 000032  
 C.FNC= 000010  
 C.LIN= 000006  
 C.LNK= 000000  
 C.MOD= 000011  
 C.NSP= 000004  
 C.PRO= 000042

C.RSV= 000002  
 C.STA= 000007  
 C.STS= 000012  
 C.URM= 177776  
 C.XACP= 000004  
 C.XID= 000035  
 C.XLEN= 000044  
 C.XPLI= 000040  
 C.XPT= 000034  
 C.XSVC= 000042  
 C.XTC= 000037  
 C.X25= 000036  
 D\$\$BUG= 177514  
 D\$\$ISK= 000000  
 D\$\$L11= 000001  
 D\$\$YNC= 000000  
 D\$\$YNM= 000000  
 E\$\$XPR= 000000  
 FC.CCP= 000020  
 FC.CTL= 000006  
 FC.KCP= 000016  
 FC.KIL= 000004  
 FC.MAN= 000024  
 FC.MLD= 000026  
 FC.PCT= 000030  
 FC.PWR= 000022  
 FC.RCE= 000002  
 FC.RCP= 000014  
 FC.TIM= 000010  
 FC.XCP= 000012  
 FC.XME= 000000  
 FS.AST= 000000  
 FS.CIB= 002000  
 FS.CRA= 001000  
 FS.DIS= 013000  
 FS.DVC= 001000  
 FS.ENR= 012000  
 FS.EXI= 001000  
 FS.GET= 006000  
 FS.HLT= 000000  
 FS.INI= 000000  
 FS.KIL= 000000  
 FS.LCL= 100000  
 FS.LTM= 001000  
 FS.MNT= 004000  
 FS.MSN= 014000  
 FS.REA= 001000  
 FS.RET= 000000  
 FS.REZ= 003000  
 FS.RLB= 002000

FS.RNG= 011000  
 FS.RST= 000000  
 FS.RTN= 001000  
 FS.SET= 005000  
 FS.SFC= 005000  
 FS.SFR= 006000  
 FS.SFS= 004000  
 FS.SPW= 040000  
 FS.STM= 000000  
 FS.STP= 002000  
 FS.STR= 001000  
 FS.TRM= 003000  
 FS.WLB= 001000  
 FS.XKL= 002000  
 FS.XOF= 010000  
 FS.XON= 007000  
 FS.ZER= 002000  
 G\$\$LVL= 000001  
 G\$\$TPP= 000000  
 G\$\$TSS= 000000  
 G\$\$TTK= 000000  
 G\$\$WRD= 000000  
 INTCT = \*\*\*\*\* GX  
 I\$\$RAR= 000000  
 I\$\$RDN= 000000  
 K\$\$CNT= 177546  
 K\$\$CSR= 177546  
 K\$\$LDC= 000000  
 K\$\$TPS= 000074  
 LD\$LP = 000000  
 L\$\$ASG= 000000  
 L\$\$DRV= 000000  
 L\$\$P11= 000001  
 L\$\$11R= 000000  
 M\$\$CRB= 000124  
 M\$\$CRX= 000000  
 M\$\$FCS= 000000  
 M\$\$MGE= 000000  
 M\$\$NET= 000000  
 M\$\$OVR= 000000  
 N\$\$ACC= 000001  
 N\$\$BUF= 000001  
 N\$\$LDV= 000001  
 N\$\$MCP= 000001  
 N\$\$MLL= 000001  
 N\$\$MOV= 000010  
 N\$\$NCT= 000001  
 N\$\$PEM= 000001  
 PDDSP = \*\*\*\*\* GX  
 PDSPL = \*\*\*\*\* GX

PDVTA = \*\*\*\*\* GX  
 PR7 = \*\*\*\*\* GX  
 PS = \*\*\*\*\* GX  
 P\$\$P45= 000000  
 P\$\$WRD= 000000  
 Q\$\$OPT= 000010  
 R\$\$DER= 000000  
 R\$\$K11= 000001  
 R\$\$SND= 000000  
 R\$\$11M= 000000  
 STDLC = \*\*\*\*\* GX  
 S\$\$WRG= 000000  
 S\$\$YSZ= 007600  
 TKPS = \*\*\*\*\* GX  
 T\$\$KMG= 000000  
 T\$\$MIN= 000000  
 V\$\$CTR= 001000  
 X\$\$DBT= 000000  
 ZF.COU= 001000  
 ZF.DDM= 000001  
 ZF.DIA= 004000  
 ZF.DLC= 000002  
 ZF.DVP= 100000  
 ZF.INI= 040000  
 ZF.KMX= 000020  
 ZF.LLC= 000004  
 ZF.LMC= 000100  
 ZF.MAN= 020000  
 ZF.MFL= 000010  
 ZF.MTM= 000400  
 ZF.MUX= 000040  
 ZF.PSE= 002000  
 ZF.SLI= 010000  
 ZF.TIP= 000200  
 ZF.X3P= 000000  
 ZS.ASN= 100000  
 ZS.BSY= 140000  
 Z.AVL= 000014  
 Z.DAT= 000016  
 Z.DSP= 000000  
 Z.FLG= 000010  
 Z.LEN= 000016  
 Z.LLN= 000006  
 Z.MAP= 000020  
 Z.NAM= 000004  
 Z.PCB= 000012  
 Z.SCH= 000007  
 \$FRKHD= \*\*\*\*\* GX  
 \$SQSRV= 000000RG  
 .\$\$\$= 000034

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
 000244 001 (RW,I,LCL,REL,CON)  
 Errors detected: 0

\*\*\* Assembler statistics



AXSUB MACRO V05.03b Friday 28-Jun-85 18:30 Page 9  
 \$DEACX - Deallocate core buffer in extended pool

```

242 .sbtll $DEACX - Deallocate core buffer in extended pool
243 +
244 **-$DEACX-Deallocate core buffer in extended pool
245
246 This routine is called to deallocate a core buffer in extended pool.
247 the block is inserted into the free block chain by core address. If
248 an adjacent block is currently free, then the two blocks are merged
249 and inserted in the free block chain.
250
251 Inputs:
252
253 R0=address of the core buffer to be deallocated.
254 R1=size of the core buffer to deallocate in bytes.
255
256 Outputs:
257
258 The core block is merged into the free core chain by core
259 address and is merged if necessary with adjacent blocks.
260
261 Registers:
262
263 R2, R3 are modified
264 -
265
266 000150 $DEACX::
267 .IF NDF M$$MGE
268
269 CALLR @DEACB ; deallocate to RSX pool in unmapped systems
270
271 .IFF
272
273 000150 032700 000001 BIT #1,R0 ; is the block in extended pool ?
274 000154 001002 BNE 5$; if ne, yes
275 000156 CALLR @DEACB ; else, deallocate RSX core block
276 000162 SAVMAP ; save current APR 6 mapping
277 000166 010446 MOV R4,-(SP) ; save some registers
278 000170 010546 MOV R5,-(SP) ; ...
279 000172 010005 MOV R0,R5 ; copy unmapped address
280 000174 010046 MOV R0,-(SP) ; set up address for $CEACC
281 000176 CALL @CEACC ; map extended address
282 000202 012604 MOV (SP)+,R4 ; copy mapped address
283 000204 SAVMAP ; save block mapping
284 000210 012703 000000' MOV #XAVLL-2,R3 ; point to allocation mask word
285 000214 061301 ADD (R3),R1 ; round to next boundary
286 000216 042301 BIC (R3)+,R1 ; clear excess
287 000220 001476 BEQ 10$; if eq, nothing to return
288 000222 017713 000000G MOV @XAVL,(R3) ; copy CEX allocation listhead
289 000226 005746 TST -(SP) ; reserve a word on the stack
290
291 000230 010316 10$: MOV R3,(SP) ; save unmapped address in reserved word
292 000232 010346 MOV R3,-(SP) ; copy address of current block
293 000234 CALL @CEACC ; map address of block
294 000240 012602 MOV (SP)+,R2 ; save mapped address
295 000242 011203 MOV (R2),R3 ; get address of next block
296 000244 001402 BEQ 20$; if eq end of chain
297 000246 020503 CMP R5,R3 ; block go before here?
298 000250 103367 BHIS 10$; if his no

```

AXTIM - AUXILLIARY PROCESS TIME MACRO V05.03b Friday 28-Jun-85 18:31<sup>H 6</sup>  
Table of contents

|     |     |                                    |
|-----|-----|------------------------------------|
| 5-  | 50  | Macro definitions                  |
| 6-  | 64  | ONCE-PER-SECOND TIMER SERVICE      |
| 8-  | 152 | PERFORM ONCE-PER-SECOND OPERATIONS |
| 9-  | 280 | 100 MSEC TIMER SERVICE             |
| 11- | 331 | SCAN SHORT TIMER QUEUE             |
| 12- | 419 | PUT TIMER ENTRY ON QUEUE           |
| 13- | 458 | START SHORT TIMER                  |

```

458 .SBTTL START SHORT TIMER
459 :+
460 **-$STSTM-START SHORT TIMER.
461 :
462 START A SHORT TIMER.
463 :
464 INPUTS:
465 R4 - POINTER TO SHORT TIMER ENTRY
466
467 000530 $STSTM::SAVRG <R0> ; GET A FREE REGISTER
468 000532 016464 000006 000004 MOV X.TMR(R4),X.TMR(R4)
469 000540 016700 000000G MOV T100Q,R0 ; GET ADDRESS OF GENERAL SHORT TIMER LISTHEAD
470
471 .IF DF R$$MPL
472 .IF NDF R$$PRO
473
474 BIT #F2.MP,@FMSK2 ;: IS THIS A MULTI-PROCESSOR?
475 BEQ 10$;: BR IF NO
476 BITB #XF.DLC!XF.DDM,X.FLAG(R4)
477 BEQ 10$;: IF EQ, USE GENERAL TIMER
478 MOV M100Q,R0 ;: USE PROCESSOR DEPENDANT LISTHEAD
479
480 10$: .ENDC
481 .ENDC
482
483 000544 011046 MOV (R0),-(SP) ; SAVE CURRENT QUEUE CONTENTS
484 000546 CALL STALT ; ADD NEW ENTRY TO THE QUEUE
485 000552 005726 TST (SP)+ ; WAS THE TIMER PREVIOUSLY ACTIVE?
486 000554 001014 BNE 40$; IF NE, YES
487
488 000556 SAVRG <R1,R2,R3,R4,R5>; SAVE ALL REGISTERS
489
490 .IF DF R$$MPL
491 .IF NDF R$$PRO
492
493 BIT #F2.MP,@FMSK2 ;: IS THIS A MULTI-PROCESSOR?
494 BEQ 20$;: BR IF NO
495 BITB #XF.DLC!XF.DDM,X.FLAG(R4)
496 BEQ 20$;: IF EQ, GENERAL TIMER
497 MOV #MP100M,C.SUB(R0); SET UP ADDRESS OF PROCESSING ROUTINE
498 MOV @TK100,R2 ; GET # OF CLOCK TICKS IN 100 MSEC
499 CALL $SMPMT ; START PROCESSOR DEPENDANT
500 BR 30$; ENTER COMMON CODE
501
502 20$: .ENDC
503 .ENDC
504
505 000570 CALL $T1001 ; INSERT TIMER ENTRY IN CLOCK QUEUE
506 000574 RESRG <R5,R4,R3,R2,R1>; RESTORE REGISTERS
507
508 000606 40$: RESRG <R0>
509 000610 RETURN
510
511 000001 .END

```

```

169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186 000300
187 000304 103422
188
189 000306 004167 000000G
190 000312 000004G
191
192 000314
193 000320 103411
194 000322 016702 000000G
195 000326 016362 000076 000006
196 000334
197 000340 000241
198 000342
199
200 000344
201
202 000352

.SBTTL Connect accept QIO processing
*
**--ACC-Connect accept request processing
This routine processes outgoing connect accept requests.
Inputs:
R1 = Address of the task's header
R2 = I/O subfunction code/4
R3 = Address of I/O packet
R4 = Address of the window block
R5 = Address of database descriptor
Outputs:
'C' Clear - Connection successfully accepted
'C' Set - Accept failed
ACC: CALL TLCHK ; Check for valid temporary link address
BCS 100$; If CS, invalid
JSR R1,CPYOPT ; Copy user optional data
.WORD I.PRM+4
CALL SNDACC ; Accept the connection
BCS 100$; If CS, resource allocation failure
MOV $WBLK,R2 ; Copy logical link segment size to window block
MOV L.SEGZ(R3),W.SEGZ(R2)
CALL IOSUC ; Complete the request successfully
CLC ; Indicate success
RETURN
10$: $IERRC IE.RSU&377 ; Return resource error
100$: $IERRC IE.BAD&377 ; Invalid temporary link address

```

SESCIL - Session control contro MACRO V05.03b Friday 28-Jun-85 19:53<sup>6 9</sup>  
Table of contents

|     |     |                                                   |
|-----|-----|---------------------------------------------------|
| 6-  | 42  | Macro definitions                                 |
| 7-  | 61  | Network control QIO processing                    |
| 8-  | 104 | Network access                                    |
| 9-  | 169 | Network deaccess                                  |
| 10- | 296 | Specify network data AST                          |
| 11- | 330 | Get network data                                  |
| 12- | 394 | Find requested mail item                          |
| 13- | 449 | Connect request network data processing           |
| 14- | 487 | Interrupt message network data processing         |
| 15- | 515 | User/network abort and disconnect data processing |
| 16- | 553 | Network event data processing                     |
| 17- | 580 | Move data to user buffer                          |
| 18- | 606 | Post event on mailbox                             |
| 19- | 700 | Access control completion                         |
| 20- | 783 | Get local node parameters                         |

```

449 .SBTTL Connect request network data processing
450 ;+
451 ;*-CON-Connect request network data processing
452 ;*-VFY-Verification request network data processing
453 ;
454 ; Convert the pending connect block into the format expected by the
455 ; user task and copy it to the user's buffer.
456 ;
457 ; Inputs:
458 ; R1 = Address of CCB
459 ; R3 = Address of I/O packet
460 ; R4 = Virtual address of user's buffer
461 ; User's buffer is mapped
462 ; R5 = Address of database descriptor
463 ; 2(SP) - # of bytes to be transferred
464 ; 4(SP) - High byte: Data type code
465 ; Low byte: I/O status return code
466 ; 6(SP) - Address of I/O packet
467 ;
468 ; Registers modified:
469 ; R0, R2, R4
470 ;
471 .PSECT
472
473 000446 CON:
474 000446 VFY: MOV 2(SP),R0 ; Get # of bytes to transfer
475 000452 01660U 000002 BEQ 10$; If EQ, none
476
477 000454 SAVRG <R1,R3> ; Save some registers
478 000460 016102 000016 MOV C.BUF+2(R1),R2 ; Get source virtual address
479 000464 005001 CLR R1 ; No addressing bias
480 000466 016303 000000G MOV I.PRM(R3),R3 ; Get destination bias
481 000472 CALL @BLXIO ; Copy the data
482 000476 RESRG <R3,R1> ; Recover registers
483
484 000502 116166 000032 000003 10$: MOVB C.FLG2(R1),3(SP); Set verification status in high byte
485 000510 RETURN

```

SESCTL11S CREATED BY MACRO ON 28-JUN-85 AT 19:54 PAGE 1 G 11  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                                                     |
|---------|------------|----------------------------------------------------------------|
| ABO     | 000540 R   | 11-389 #15-539                                                 |
| ABT     | 000550 R   | 11-388 #15-542                                                 |
| ACCLLT  | = ***** GX | 9-232                                                          |
| ADDGNO  | = ***** GX | 18-688                                                         |
| ADDMAI  | = ***** GX | 18-691                                                         |
| BADERR  | 001224 R   | 19-723 #19-777                                                 |
| BLXIO   | = ***** GX | 13-481 17-602 18-654 19-738                                    |
| CLS     | 000000 R   | 7-90 #9-187                                                    |
| CLSDON  | 000176 R   | 9-195 #9-258                                                   |
| CLSTA   | 000266 R   | 9-235 #9-289                                                   |
| CMPINT  | = ***** GX | 14-513                                                         |
| CM.CON  | = 000200   | 8-137 19-743                                                   |
| CON     | 000446 R   | 11-385 #13-473                                                 |
| CTLDSP  | 000004 R   | 7-83 #7-89                                                     |
| CX.SMC  | = 000010   | 18-683                                                         |
| CX.UNL  | = 000004   | 10-321 11-365 12-424 19-744                                    |
| C.BUF   | 000014     | 13-478 17-595 17-600 18-652 18-653 18-660 18-661 19-734 19-735 |
| C.CNT   | 000020     | 11-353 *18-648                                                 |
| C.CNT2  | 000030     | 18-663                                                         |
| C.FLG2  | 000032     | 13-484 *19-727 19-748 19-760                                   |
| C.FNC   | 000010     | 8-137 8-144 10-318 11-363 12-422 *18-683 *19-743               |
| C.MOD   | 000011     | 10-321 *11-365 12-424 12-439 14-511 15-548 *19-744             |
| C.NSP   | 000004     | 8-130 15-539                                                   |
| C.STS   | 000012     | *18-684                                                        |
| DEACB   | = ***** GX | 9-282                                                          |
| DECP    | = ***** GX | 20-801 20-816                                                  |
| DISCMP  | = ***** GX | 15-549                                                         |
| DSC     | 000550 R   | 11-387 #15-543                                                 |
| D\$LNAM | 000006     | 20-802                                                         |
| D\$LNUM | 000014     | 20-822                                                         |
| D\$SEG  | 000036     | 20-817 20-818                                                  |
| ER\$ABO | = 000046   | 9-234                                                          |
| ER\$ACC | = 000042   | 19-745                                                         |
| ER\$MLB | = 000006   | 8-156                                                          |
| ER\$UOB | = 000004   | 19-766                                                         |
| EVT     | 000576 R   | 11-390 #16-573                                                 |
| FLSHIO  | = ***** GX | 9-225                                                          |
| FLSHMB  | = ***** GX | 9-192                                                          |
| FNDMAI  | 000244 R   | 11-346 #12-418                                                 |
| FNDMBX  | = ***** GX | 18-685                                                         |
| GLN     | 001232 R   | 7-102 #20-799                                                  |
| GND     | 000302 R   | 7-92 7-93 7-94 #11-346                                         |
| H.LUN   | = ***** GX | 9-206                                                          |
| H.NLUN  | = ***** GX | 9-205                                                          |
| H.NML   | = ***** GX | *9-278                                                         |
| IE.ABO  | = ***** GX | 9-228                                                          |
| IE.BAD  | = ***** GX | 19-777                                                         |
| IE.INS  | = ***** GX | 18-695                                                         |
| IE.NDA  | = ***** GX | 11-379                                                         |
| IE.RSU  | = ***** GX | 18-645                                                         |
| IE.SPC  | = ***** GX | 18-641                                                         |

6 12

SE\$CTR - Session control counte MACRO V05.03b Friday 28-Jun-85 19:54 Page 9-3  
 Symbol table

|                 |                |                |                 |                    |
|-----------------|----------------|----------------|-----------------|--------------------|
| S\$SYSZ= 007600 | T\$LLDC 000045 | T\$LTPS 000020 | T\$NVR 000001   | US\$DON= 000000    |
| T\$FLAG 000044  | T\$LLDL 000012 | T\$NAPL 000004 | T\$RPRI 000040  | US\$DSC= 000004    |
| T\$LIF 000013   | T\$LLDO 000012 | T\$NFE 000000  | T\$SVC 000034   | US\$EAC= 000012    |
| T\$LIFL 000013  | T\$LLDS 000012 | T\$NLEN 000010 | T\$T5 000030    | US\$WDS= 000010    |
| T\$LIFO 000013  | T\$LLDN 000046 | T\$NNUL 000002 | T\$T6 000032    | V\$SCTR= 001000    |
| T\$LIFS 000013  | T\$LOPR 000002 | T\$NOPL 000006 | T\$KMG= 000000  | X\$DBT= 000000     |
| T\$LIN 000000   | T\$LTCL 000024 | T\$NRNJ 000042 | T\$MIN= 000000  | \$BYTE = ***** GX  |
| T\$LIPS 000006  | T\$LTJM 000026 | T\$NRPL 000005 | US\$CNF= 000002 | \$S\$FT= 000001    |
| T\$LLD 000012   | T\$LTPR 000014 | T\$NRUL 000007 | US\$DIS= 000006 | .\$\$\$\$.= 000034 |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
 000242 001 (RW,I,LCL,REL,CON)  
 Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 107  
 Work file writes: 102  
 Size of work file: 26175 Words ( 103 Pages)  
 Size of core pool: 17608 Words ( 67 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:23.41  
 SY:SE\$CTR11S.V2,[131,134]SE\$CTR11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCS/PA:1,[131,10]V2,SE\$CTR



Local buffers

```

108 .SBTTL Local buffers
109 ;+
110 ; Note that the buffers defined below must remain in the same order.
111 ; -
112
113 000070 $CNBLK:: ; Buffer for current outgoing connect request
114 000070 $DSNOD::.BLKB 6 ; Destination node address
115 000076 $DSDFM::.BLKB 1 ; Destination descriptor format
116 000077 $DSOBJ::.BLKB 1 ; Destination object type
117 000100 $WORK:: ; Workspace area
118 000100 $DSDSC::.BLKB 18. ; Destination process descriptor
119 000122 $SRDFM::.BLKB 1 ; Source descriptor format
120 000123 $SROBJ::.BLKB 1 ; Source object type
121 000124 $SRDSC::.BLKB 18. ; Source process descriptor
122 000146 $REQID::.BLKB 18. ; Requestor ID
123 000170 $PASSW::.BLKB 10. ; Password
124 000202 $ACCN1::.BLKB 18. ; Accounting info
125 000224 $OPLNG::.WORD 0 ; Optional data length
126 000226 $OPDAT::.BLKB 16. ; Optional data buffer
000000

```

```

125 .SBTTL Connect reject QIO processing
126
127 + **--REJ--Connect reject QIO processing
128
129 This routine is called to reject an incoming connect request.
130
131 - Inputs:
132 R1 = Address of task's header
133 R3 = Address of I/O packet
134 R4 = Address of mailbox
135 R5 = Address of database descriptor
136
137 000076 010467 000000G REJ: MOV R4,$MAIBX ; Set up mailbox address
138 000102 CALL TLACHK ; Check for valid temporary link address
139 000106 103414 BCS 100$; If CS, invalid
140
141 000110 105364 000010 DEC M,USE(R4) ; Reduce count of active/pending links
142 000114 004167 000000G JSR R1,CPYOPT ; Copy optional data to LLT
143 000120 000004G .WORD I,PRM+4
144 000122 CALL SAVOPT ; and save it in the LLT
145
146 000126 005001 CLR R1 ; Reason = reject by user
147 000130 CALL BRKLNK ; Break the logical link
148
149 000134 CALLR IOSUC ; Complete the I/O request
150
151 000140 100$: $IERRC IE.BAD&377 ; Invalid temporary link address
152
153 .END
000001

```

SESDMO11S CREATED BY MACRO ON 28-JUN-85 AT 19:55

PAGE 1 6 15

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                               |
|---------|------------|------------------------------------------|
| ACCLLT  | = ***** GX | 8-126                                    |
| AC\$DNT | = 000002   | #6-48                                    |
| AC\$X25 | = 000001   | #6-48                                    |
| AE\$CIR | = 000003   | #6-48                                    |
| AE\$LIN | = 000001   | #6-48                                    |
| AE\$MOD | = 000004   | #6-48                                    |
| BYTE3   | = 000300   | #7-74 7-74 #7-74 7-74 7-74 7-74 7-74     |
| CL\$ASZ | = 010500   | #6-48                                    |
| CL\$DLL | = 000500   | #6-48 6-48 6-48 6-48 6-48 6-48 6-48 6-48 |
| CL\$ECL | = 000300   | #6-48                                    |
| CL\$LDN | = 010400   | #6-48                                    |
| CL\$MAN | = 000000   | #6-48 6-48 6-48                          |
| CL\$PAZ | = 034100   | #6-48 6-48 6-48                          |
| CL\$PLH | = 034000   | #6-48 6-48 6-48                          |
| CL\$PLL | = 000600   | #6-48                                    |
| CL\$PRT | = 034200   | #6-48                                    |
| CL\$ROU | = 010000   | #6-48 6-48 6-48                          |
| CL\$SES | = 000200   | #6-48 6-48 6-48                          |
| CL\$SGE | = 035000   | #6-48 6-48 6-48                          |
| CL\$SSE | = 035100   | #6-48 6-48 6-48 6-48 6-48 6-48 6-48 6-48 |
| CL\$TRN | = 000400   | #6-48 6-48 6-48 6-48 6-48 6-48 6-48 6-48 |
| CL\$XL2 | = 013700   | #6-48                                    |
| CL\$XL3 | = 013600   | #6-48 6-48                               |
| CL\$X2S | = 013500   | #6-48 6-48                               |
| DL\$AST | = 000002   | #6-48                                    |
| DL\$HLT | = 000000   | #6-48                                    |
| DL\$IST | = 000001   | #6-48                                    |
| DL\$MAI | = 000004   | #6-48                                    |
| DL\$OFF | = 000001   | #6-48                                    |
| DL\$ON  | = 000000   | #6-48                                    |
| DL\$RUN | = 000003   | #6-48                                    |
| DL\$SHU | = 000002   | #6-48                                    |
| DL\$SYN | = 000005   | #6-48                                    |
| DMOTA   | = 000062 R | 8-130 #8-141                             |
| EF\$ACT | = 000001   | #6-48                                    |
| ER\$NSL | = 000013   | 8-129                                    |
| ER\$NSR | = 000003   | 7-98                                     |
| EVL\$ES | = ***** GX | 7-74                                     |
| EV\$ACF | = 000201   | #6-48                                    |
| EV\$ADR | = 000420   | #6-48                                    |
| EV\$ADU | = 000417   | #6-48                                    |
| EV\$APL | = 000400   | #6-48                                    |
| EV\$ARC | = 000421   | #6-48                                    |
| EV\$AUC | = 000010   | #6-48                                    |
| EV\$AUS | = 000003   | #6-48                                    |
| EV\$CDF | = 000520   | #6-48                                    |
| EV\$COZ | = 000011   | #6-48                                    |
| EV\$DBR | = 000302   | #6-48                                    |
| EV\$GAS | = 035101   | #6-48                                    |

SESDSP - Session control dispat MACRO V05.03b Friday 28-Jun-85 19:56 Page 8-2  
Session control ACP idle loop

G.16

```
212
213
214
215
216
217
218
219
220 000312
221

 MOV $UCB,R5 ;; Get address of NS: UCB
 CLR U.ACP(R5) ;; Mark ACP inactive
 MOV @TKTCB,R5 ;; Get address of my TCB
 CALLR @DREXT ;; and exit

951: .ENDC

 CALLR @STPCT ;; Stop ACP until we have some work to do
```

H.16

|     |     |                                                 |
|-----|-----|-------------------------------------------------|
| 5-  | 56  | Macro definitions                               |
| 6-  | 106 | TIMER SERVICE - STATE DISPATCH TABLE            |
| 7-  | 129 | CONTROL ENABLE SERVICE                          |
| 8-  | 166 | LOOK FOR RING                                   |
| 9-  | 184 | ENABLE LINE                                     |
| 10- | 222 | DISABLE LINE                                    |
| 11- | 259 | POST ABORT COMPLETION TO DLC LEVEL              |
| 12- | 284 | MODEM CONTROLLER INTERRUPT ROUTINE              |
| 13- | 313 | MODEM CONTROLLER TIMER SERVICE                  |
| 14- | 379 | ST.WRG - WAIT FOR RING DETECT                   |
| 15- | 399 | ST.WCN - WAIT FOR CONNECT DETECT                |
| 16- | 432 | ST.CDL - CONNECT DELAY                          |
| 17- | 502 | CNCMP - COMPLETE THE CONNECTION                 |
| 18- | 522 | ST.ACT - LINE ACTIVE                            |
| 19- | 568 | ST.ADL - ACTIVE DELAY                           |
| 20- | 590 | POST ASYNCHRONOUS DISCONNECT                    |
| 21- | 610 | ST.CER - CONNECT ERROR                          |
| 22- | 630 | ST.ABO - POST ABORT COMPLETION TO THE DLC LEVEL |
| 23- | 651 | ST.DDL - DISCONNECT DELAY                       |
| 24- | 673 | SCHEDULE DLC PROCESS                            |

```

399 .SBTTL ST.WCN - WAIT FOR CONNECT DETECT
400
401 +
402 ***.WCN-WAIT FOR CONNECT DETECT
403
404 WAIT UNTIL CARRIER (ASYNCR) OR DATASET READY (SYNCR) HAS BEEN ASSERTED.
405
406 INPUTS:
407 R2 - POINTER TO SYSTEM LINE TABLE
408 R4 - SYSTEM LINE NUMBER
409 R5 - POINTER TO LINE ENTRY IN MDC DATABASE
410
411 .WCN: BIT #LF.LPB,(R2) ; IS THIS LINE IN LOOPBACK?
412 BNE CNCRMP ; IF NE, YES
413 .IF NDF X$$D52
414 BITB #MS.SYN,(R5) ; IS THIS A SYNC LINE?
415 BNE 20$; YES ... CHECK FOR DATASET READY
416 BITB #MC.CAR,M.CSV(R5) ; NO ... IS CARRIER ASSERTED?
417 BEQ 10$; NO ... KEEP WAITING
418 MOVB #TM.CON,M.TIM(R5) ; YES ... SET UP TIMEOUT CELL
419 MOVB #ST.CDL,M.STT(R5) ; SET UP NEW STATE
420 10$: RETURN ; AND RETURN
421 20$: BITB #MC.DSR,M.CSV(R5) ; SYNC LINE...IS DATASET READY ASSERTED?
422 BNE CNCRMP ; YES ... COMPLETE THE CONNECT
423 RETURN
424
425 ;
426 ; Wait until Data Set Ready has been asserted.
427
428 BITB #MC.DSR, M.CSV(R5) ; If Data Set Ready is asserted
429 BEQ 10$; then
430 BITB #MS.SWT, M.STS(R5) ; If the modem switches Carrier
431 BNE CNCRMP ; then goto complete the connect.
432 MOVB #ST.CDL, M.STT(R5) ; else Set state to connect delay
433 MOVB #TM.CHK, M.TIM(R5) ; and start timer.
434
435 10$: RETURN
436 .ENDC

```

AXMDC CREATED BY MACRO ON 28-JUN-85 AT 18:30 PAGE 1 H 3  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL | VALUE      | REFERENCES                   |
|--------|------------|------------------------------|
| CCBGT  | = ***** GX | 24-692                       |
| CE.ABO | = 100362   | 11-275                       |
| CE.DIS | = 100366   | 20-603                       |
| CNCMP  | = 000610 R | 15-411 15-421 16-454 #17-513 |
| CS.DIS | = 000040   | 23-664                       |
| CS.ENB | = 000020   | 17-513                       |
| CS.FTL | = 001000   | 20-603                       |
| CS.RNG | = 000010   | 14-392                       |
| CTLDSP | = 000060 R | 7-156 #7-161                 |
| C.FNC  | = 000010   | *24-695                      |
| C.LIN  | = 000006   | 7-150 *24-696                |
| C.STA  | = 000007   | 7-155 *24-697                |
| DDAST  | = ***** GX | 20-604                       |
| DDCCP  | = ***** GX | 24-698                       |
| DDMSN  | = ***** GX | 13-361                       |
| DISC   | = 000756 R | 18-547 19-584 #20-602 21-627 |
| DSABLE | = 000134 R | 7-163 #10-236                |
| ENABLE | = 000114 R | 7-162 #9-194                 |
| FS.DIS | = 013000   | 23-666                       |
| FS.ENB | = 012000   | 11-277 17-515                |
| FS.RNG | = 011000   | 7-156 14-394                 |
| IS\$AS | = *****    | 5-63 12-283                  |
| LF.ACT | = 100000   | #5-66                        |
| LF.BRO | = 000400   | #5-66                        |
| LF.BWT | = 000007   | #5-66                        |
| LF.ENA | = 002000   | #5-66                        |
| LF.LPB | = 001000   | 15-410 18-535                |
| LF.MDC | = 000100   | #5-66                        |
| LF.MFL | = 004000   | #5-66                        |
| LF.MTP | = 000020   | #5-66                        |
| LF.PAC | = 000200   | #5-66                        |
| LF.RDY | = 040000   | #5-66                        |
| LF.REA | = 010000   | #5-66                        |
| LF.SER | = 000040   | #5-66                        |
| LF.TIM | = 000010   | #5-66                        |
| LF.UNL | = 020000   | #5-66                        |
| LF.X2P | = 000000   | #5-66                        |
| LN.CLO | = 000000   | #5-66                        |
| LN.DUM | = 000005   | #5-66                        |
| LN.LOA | = 000004   | #5-66                        |
| LN.LOO | = 000003   | #5-66                        |
| LN.OAU | = 000003   | #5-66                        |
| LN.OFF | = 000001   | #5-66                        |
| LN.ON  | = 000000   | #5-66                        |
| LN.OOP | = 000004   | #5-66                        |
| LN.OPE | = 000001   | #5-66                        |
| LN.REF | = 000002   | #5-66                        |
| LN.SER | = 000002   | #5-66                        |
| LN.STA | = 000017   | #5-66                        |
| LN.SUB | = 000360   | #5-66                        |
| LN.TRI | = 000006   | #5-66                        |
| L.COST | = 000015   | #5-66                        |

AXSCH MACRO V05.03b Friday 28-Jun-85 18:30 Page 6-4  
Symbol table

H 4

Work file reads: 0  
Work file writes: 0  
Size of work file: 15916 Words ( 63 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:11.32

SY:AXSCH.V2,[131,134]AXSCH/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]AXSCH



```

299
300 000252 20$: SAVMAP ; save current mapping
301 000256 MAP 4(SP) ; map block being released
302 000264 010314 MOV R3,(R4) ; assume blocks not adjacent
303 000268 RESMAP ; restore mapping
304 000272 010546 MOV R5,-(SP) ; calculate address of new block
305 000274 060116 ADD R1,(SP) ;
306 000276 020326 CMP R3,(SP)+ ; equal to next in chain?
307 000300 001016 BNE 30$; if ne no
308 000302 SAVMAP ; else, save mapping
309 000306 010346 MOV R3,-(SP) ; copy block address
310 000310 CALL @CEACC ; map block
311 000314 012600 MOV (SP)+,R0 ; get mapped address
312 000316 012046 MOV (R0)+,-(SP) ; save link word
313 000320 061001 ADD (R0),R1 ; merge two blocks
314 000322 MAP 6(SP) ; map block being released
315 000330 012614 MOV (SP)+,(R4) ; move link word to block released
316 000332 RESMAP ; restore mapping
317 000336 30$: ; address of previous block is on top of stack
318 000336 010522 MOV R5,(R2)+ ; assume no agglomeration
319 000340 061216 ADD (R2),(SP) ; calculate address of next block
320 000342 020526 CMP R5,(SP)+ ; equal to block being released?
321 000344 001015 BNE 40$; if ne no
322 000346 061201 ADD (R2),R1 ; merge two blocks
323 000350 SAVMAP ; save current mapping
324 000354 MAP 2(SP) ; map to block being released
325 000362 011446 MOV (R4)+,-(SP) ; save link word
326 000364 MAP 2(SP) ; restore mapping
327 000372 012642 MOV (SP)+,-(R2) ; move link word to previous block
328 000374 012616 MOV (SP)+,(SP) ; replace block mapping
329 000376 010204 MOV R2,R4 ; set new address of block
330 000400 MAP (SP) ; map to block released
331 000404 010164 000002 40$: MOV R1,2(R4) ; set size of block released
332 000410 016777 177366 000000G MOV XAVLL,@XAVL ; update CEX copy of listhead
333
334 000416 005726 100$: TST (SP)+ ; clean up stack
335 000420 012605 MOV (SP)+,R5 ; restore registers
336 000422 012604 MOV (SP)+,R4 ;
337 000424 RESMAP ;
338 000430 RETURN ; restore APR 6 mapping
339 ; and return
340 .ENDC

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48

```

: TITLE AXTIM - AUXILLIARY PROCESS TIMER SUPPORT
: IDENT /V05.00/

: COPYRIGHT (C) 1980, 1982, 1983, 1985 BY
: DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

: THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
: SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
: INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR
: ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE
: MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH
: SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE
: TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN
: IN DEC.

: THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
: NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
: EQUIPMENT CORPORATION.

: DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
: ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

: MODULE DESCRIPTION

: CEX TIMER SERVICE ROUTINES (RESIDENT IN AUX)

: DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

: IDENT HISTORY:

: 2.00 22-OCT-80
: RE-WRITE FOR NEW MULTI-PROCESSOR TIMER SUPPORT

: 3.00 16-APR-82
: DECNET-11M V3.1
: DECNET-11M-PLUS V1.1

: 4.00 07-NOV-83
: DECNET-11M V4.0
: DECNET-11M-PLUS V2.0

: 5.00 22-JUL-85
: DECnet-11M/S V4.2
: DECnet-11M-Plus V3.0
: DECnet-Micro/RSX V1.0

```

|                  |                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|------------------|
| A\$\$CHK= 000000 | I\$\$RDN= 000000 | L.CVA 177776     | R\$\$DER= 000000 | XF.LLC= 000004   |
| A\$\$CPS= 000000 | K\$\$CNT= 177546 | L.DDM 000002     | R\$\$K11= 000001 | X\$\$DBT= 000000 |
| A\$\$PRI= 000000 | K\$\$CSR= 177546 | L.DDS 000004     | R\$\$SND= 000000 | X.FLAG 000003    |
| A\$\$TRP= 000000 | K\$\$LDC= 000000 | L.DLC 000003     | R\$\$TIM= 000000 | X.ID 000002      |
| CLCINS 000042R   | K\$\$TPS= 000074 | L.DLM 000006     | SCANST 000300R   | X.LINK 000000    |
| CLINS = ***** GX | LD\$LP= 000000   | L.DLS 000010     | SF.ACT= 000200   | X.RTMR 000006    |
| C\$\$CKP= 000000 | LF.ACT= 100000   | L.FLG 000000     | SF.ENA= 000100   | X.TMR 000004     |
| C\$\$ORE= 000400 | LF.BRO= 000400   | L.KRBA 000016    | SF.LPB= 000004   | ZF.COU= 001000   |
| C\$\$RSH= 177564 | LF.BWT= 000007   | L.LEN = 000022   | SF.MFL= 000040   | ZF.DDM= 000001   |
| C.ARS 000014     | LF.ENA= 002000   | L.MPF 000022     | SF.PAC= 000020   | ZF.DIA= 004000   |
| C.AST 000012     | LF.LPB= 001000   | L.NMST 000020    | SF.REA= 000010   | ZF.DLC= 000002   |
| C.CSTP= 000012   | LF.MDC= 000100   | L.NSTA 000014    | SF.SER= 000001   | ZF.DVP= 100000   |
| C.DST 000016     | LF.MFL= 004000   | L.OWNR 000021    | SF.SVC= 000002   | ZF.INJ= 040000   |
| C.EFN 000003     | LF.MTP= 000020   | L.UNT 000013     | SF.UNL= 000040   | ZF.KMX= 000020   |
| C.LGTH= 000020   | LF.PAC= 000200   | M\$\$CRB= 000124 | SLTMA = ***** GX | ZF.LLC= 000004   |
| C.LNK 000000     | LF.RDY= 040000   | M\$\$CRX= 000000 | SLTNM = ***** GX | ZF.LMC= 000100   |
| C.MRKT= 000000   | LF.REA= 010000   | M\$\$FCS= 000000 | STALT 000476R    | ZF.MAN= 020000   |
| C.RQT 000002     | LF.SER= 000040   | M\$\$MGE= 000000 | STDD1 = ***** GX | ZF.MFL= 000010   |
| C.RSI 000012     | LF.TIM= 000010   | M\$\$NET= 000000 | STD11 = ***** GX | ZF.MTM= 000400   |
| C.SCHD= 000002   | LF.UNL= 020000   | M\$\$OVR= 000000 | STMFC = ***** GX | ZF.MUX= 000040   |
| C.SRC 000014     | LF.X2P= 000000   | N\$\$ACC= 000001 | S\$\$WRG= 000000 | ZF.PSE= 002000   |
| C.SSHT= 000004   | LN.CLO= 000000   | N\$\$BUC= 000001 | S\$\$YSZ= 007600 | ZF.SLI= 010000   |
| C.SUB 000012     | LN.DUM= 000005   | N\$\$LDV= 000001 | S.COST 000001    | ZF.TIM= 000200   |
| C.SYST= 000006   | LN.LOA= 000004   | N\$\$MCP= 000001 | S.FLG 000000     | ZF.X3P= 000000   |
| C.SYTK= 000010   | LN.LOD= 000003   | N\$\$MLL= 000001 | S.LEN 000004     | ZS.ASN= 100000   |
| C.TCB 000004     | LN.OAU= 000003   | N\$\$MOV= 000010 | S.NMST 000002    | ZS.BSY= 140000   |
| C.TIM 000006     | LN.OFF= 000001   | N\$\$NCT= 000001 | S.OWNR 000003    | ZTIM2 = ***** GX |
| C.UIC 000016     | LN.ON = 000000   | N\$\$PEM= 000001 | TIMFLG 000054R   | Z.AVL 000014     |
| DSPTM = ***** GX | LN.OOP= 000004   | ONESEC 000056R   | TKPS = ***** GX  | Z.DAT 000016     |
| D\$\$BUG= 177514 | LN.OPE= 000001   | PDDSP = ***** GX | TK100 = ***** GX | Z.DSP 000000     |
| D\$\$ISK= 000000 | LN.OPE= 000001   | PDSPL = ***** GX | TSTIM = ***** GX | Z.FLG 000010     |
| D\$\$L11= 000001 | LN.REF= 000002   | PDVNM = ***** GX | T\$KMG= 000000   | Z.LEN = 000016   |
| D\$\$YNC= 000000 | LN.SER= 000002   | PDVTA = ***** GX | T\$MIN= 000000   | Z.LLN 000006     |
| D\$\$YNM= 000000 | LN.STA= 000017   | PR1SC 000000RG   | T1SCL = ***** GX | Z.MAP 000020     |
| E\$XPR= 000000   | LN.SUB= 000360   | PR100M 000244RG  | T100C = ***** GX | Z.NAM 000004     |
| F\$LVL= 000001   | LN.TRI= 000006   | PR7 = ***** GX   | T100Q = ***** GX | Z.PCB 000012     |
| G\$STPP= 000000  | L\$ASG= 000000   | PS = ***** GX    | V\$CTR= 001000   | Z.SCH 000007     |
| G\$STSS= 000000  | L\$DRV= 000000   | PWF1 = ***** GX  | XF.CAN= 000200   | \$STSTM 000530RG |
| G\$STTK= 000000  | L\$P11= 000001   | P\$P45= 000000   | XF.DDM= 000001   | \$TISIN 000032RG |
| G\$WRD= 000000   | L\$T1R= 000000   | P\$WRD= 000000   | XF.DLC= 000002   | \$T1001 000254R  |
| I\$RAR= 000000   | L.COST 000015    | Q\$OPT= 000010   |                  |                  |
|                  | L.CTL 000012     |                  |                  |                  |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000612 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 12660 Words ( 50 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:15.11

SY:AXTIM.V2,[131,134]AXTIM/CR/-SP=SY:[1,1]RSXCMC.SML/ML,[130,110]NETLIB/ML,[130,10]RSXCMC/PA:1,[131,10]AXTIM

```

204 .SBTTL Process incoming connect request
205
206 **--USRCI-Process incoming connect request
207
208 Process a received connect initiate message.
209
210 Inputs:
211 R3 = Virtual address of LLT
212 R4 = Address of connect pending CCB
213 R5 = Address of database descriptor
214
215 Outputs:
216 'C' Clear - Request successfully processed
217 'C' Set - Error processing request
218 N$ERRC(R5) = Reject reason code
219
220 Registers modified:
221 R0, R1, R2, R4
222
223 USRCI:: SAVRG <R3> ; Get a free register
224 000360 012765 000003 000022 MOV #RNSR,NERRC(R5) ; Set up 'node shutting down' error
225 000370 132765 000010 000005 BITB #NDMO,NFLG(R5); Is this node shutting down?
226 000376 001144 BNE 100$; If NE, yes ... disallow incoming connects
227 000400 017700 000003G MOV @DECT,R0 ; Point to the DEC home block
228 000404 026064 000014 000032 CMP D$ENUM(R0),C.FLG2(R4) ; Is the connect from ourselves
229 000412 001404 BCF 5$; If EQ, yes, don't restrict from ourselves
230 000414 132765 000002 000005 BITB #N$FRST,N$FLG(R5); Is this node in restricted state?
231 000422 001132 BNE 100$; If NE, yes, return 'node shutting down'
232 000424 012765 000004 000022 5$: MOV #R$UDB,N$ERRC(R5)
233
234 .IF DF N$MCP
235
236 MOV #1,$RQCPY ; Initialise number of copies
237
238 .ENDC
239
240 000432 016400 000016 MOV C.BUF+2(R4),R0 ; Get address of connect pending block
241 000436 105760 000005 TSTB N.DOT(R0) ; Is this a connect by object type
242 000442 001022 BNE 20$; If NE, yes
243 000444 126027 000004 000002 CMPB N.DFM(R0),#2 ; Is this a format 2 descriptor?
244 000452 001001 BNE 10$; If NE, no
245 000454 022020 CMP (R0)+,(R0)+ ; Skip over group and user
246
247 000456 062700 000010 10$: ADD #N.DDE,R0 ; Point to task name in ascii
248 000462 005067 000002G CLR $RQNAM+2 ; In case task name < 3 characters long
249 000466 CALL CAT5 ; Convert ascii to RAD50
250 000472 010167 000000G MOV R1,$RQNAM ; Fill in first 3 chars of task name
251 000476 103404 BCS 20$; If CS, task name finished
252 000500 CALL CAT5 ; Convert ascii to RAD50
253 000504 010167 000002G MOV R1,$RQNAM+2 ; Fill in second 3 chars of task name
254
255 000510 016403 000016 20$: MOV C.BUF+2(R4),R3 ; Recover pointer to pending connect block
256 000514 116301 000005 MOVB N.DOT(R3),R1 ; Get destination object type
257 000520 CALL MAPOBJ ; Map object type to task name
258 000524 103471 BCS 100$; If CS, no mapping present
259 000526 016743 000002G MOV $RQNAM+2,-(R3) ; Fill in the destination task name
260 000532 016743 000000G MOV $RQNAM,-(R3) ; ...

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

.TITLE SECTL - Session control control qio processing  
.IDENT /V05.00/  
.ENABL LC

Copyright (C) 1982, 1983, 1985 by  
Digital Equipment Corporation, Maynard, MASS.

This software is furnished under a license for use only on a  
single computer system and may be copied only with the  
inclusion of the above copyright notice. This software, or  
any other copies thereof, may not be provided or otherwise  
made available to any other person except for use on such  
system and to one who agrees to these license terms. Title  
to and ownership of the software shall at all times remain  
in DEC.

The information in this document is subject to change without  
notice and should not be construed as a commitment by Digital  
Equipment Corporation.

DEC assumes no responsibility for the use or reliability of  
its software on equipment which is not supplied by DEC.

#### Module description

Session control control QIO processing

#### Ident history:

- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

```

487 .SBTTL Interrupt message network data processing
488 ;+
489 ;**--INT-Interrupt message network data processing
490 ;
491 ; Copy the interrupt message into the user's buffer.
492 ;
493 ; Inputs:
494 ; R1 = Address of CCB
495 ; R3 = Address of I/O packet
496 ; R4 = Virtual address of user's buffer
497 ; User's buffer is mapped
498 ; R5 = Address of database descriptor
499 ; 2(SP) - # of bytes to be transferred
500 ; 4(SP) - High byte: Data type code
501 ; Low byte: I/O status return code
502 ; 6(SP) - Address of I/O packet
503 ;
504 ; Registers modified:
505 ; R0, R2, R4
506
507 000512 016600 000002 INT: MOV 2(SP),R0 ; Get # of bytes to transfer
508 000516 001402 BEQ 10$; If EQ, none
509
510 000520 CALL MOVDAT ; Copy interrupt message to user buffer
511 000524 116166 000011 000003 10$: MOV C,MOD(R1),3(SP) ; Put LUN in high byte of second I/O status word
512 000532 010104 MOV R1,R4 ; Copy the CCB address
513 000534 CALLR CMPINT ; Return interrupt message to system

```

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES |         |         |         |         |        |         |         |        |  |
|---------|------------|------------|---------|---------|---------|---------|--------|---------|---------|--------|--|
| INT     | 000512 R   | 11-386     | #14-507 |         |         |         |        |         |         |        |  |
| IODUN   | = ***** GX | 10-328     | 11-377  | 20-826  |         |         |        |         |         |        |  |
| IODUN1  | = ***** GX | 9-229      |         |         |         |         |        |         |         |        |  |
| IOERR   | = ***** GX | 11-379     | 18-641  | 18-645  | 18-695  | 19-777  |        |         |         |        |  |
| IOSUC   | = ***** GX | 8-167      | 9-283   | 18-692  | 19-740  |         |        |         |         |        |  |
| IS.DAO  | = ***** GX | 11-361     | 20-812  |         |         |         |        |         |         |        |  |
| IS.SUC  | = ***** GX | 10-327     | 11-351  | 20-824  |         |         |        |         |         |        |  |
| I.LN2   | = ***** GX | 9-277      |         |         |         |         |        |         |         |        |  |
| I.PRM   | = ***** GX | 10-312     | 11-358  | 11-360  | 11-370  | 11-371  | 12-432 | 12-434  | 12-437  | 12-439 |  |
|         |            | 13-480     | 17-601  | 18-638  | 18-649  | 18-650  | 19-717 | 19-718  | *19-719 | 19-727 |  |
|         |            | 19-731     | 19-732  | 20-799  | 20-800  | 20-804  | 20-806 | *20-808 | 20-813  | 20-820 |  |
|         |            | 20-825     |         |         |         |         |        |         |         |        |  |
| I.TCB   | = ***** GX | 18-624     |         |         |         |         |        |         |         |        |  |
| KILLNK  | = ***** GX | 9-236      |         |         |         |         |        |         |         |        |  |
| KISAR6  | = ***** GX | *11-370    | *18-660 | *18-682 | *19-717 | *20-799 |        |         |         |        |  |
| LDBG1   | = ***** GX | 18-643     |         |         |         |         |        |         |         |        |  |
| LDBRT   | = ***** GX | 18-694     |         |         |         |         |        |         |         |        |  |
| L.DCR   | 000100     | *9-234     |         |         |         |         |        |         |         |        |  |
| MAPBF   | 000662 R   | 18-656     | #18-660 |         |         |         |        |         |         |        |  |
| MOV DAT | 000616 R   | 14-510     | 15-545  | 16-576  | #17-594 |         |        |         |         |        |  |
| M.MAIL  | 000014     | 8-124      | 10-313  | 12-418  |         |         |        |         |         |        |  |
| M.MAX   | 000011     | 8-149      | 8-151   |         |         |         |        |         |         |        |  |
| M.MBL   | 000020     | 9-281      |         |         |         |         |        |         |         |        |  |
| M.NEXT  | 000002     | *8-117     | 9-259   | 9-261   | 9-263   | 9-266   | *9-266 |         |         |        |  |
| M.SPA   | 000012     | *9-187     | 9-267   | *10-312 |         |         |        |         |         |        |  |
| M.TASK  | 000004     | 8-130      |         |         |         |         |        |         |         |        |  |
| M.USE   | 000010     | 8-151      | *8-161  | 9-194   | *9-241  | *19-725 |        |         |         |        |  |
| NS\$DON | = 000000   | 9-289      |         |         |         |         |        |         |         |        |  |
| NS\$SDI | = 000002   | 9-290      | 9-292   | 9-294   |         |         |        |         |         |        |  |
| NT.CON  | = 000001   | 19-743     |         |         |         |         |        |         |         |        |  |
| NT.EVT  | = 000006   | 8-144      | 18-683  |         |         |         |        |         |         |        |  |
| NT.VFY  | = 000007   | 8-137      |         |         |         |         |        |         |         |        |  |
| NVP     | 001006 R   | 18-632     | #19-717 |         |         |         |        |         |         |        |  |
| NSERRC  | 000022     | *19-745    | *19-766 | 19-772  |         |         |        |         |         |        |  |
| NSGENQ  | 000052     | 8-121      |         |         |         |         |        |         |         |        |  |
| NSMBXQ  | 000050     | 8-117      | *8-118  | 9-259   |         |         |        |         |         |        |  |
| NS\$ACC | = 000001   | 7-96       | 8-135   | 18-620  | 18-628  | 19-715  |        |         |         |        |  |
| NS\$EVL | = 000001   | #4-2       |         |         |         |         |        |         |         |        |  |
| NS\$PEM | = 000001   | 7-96       | 8-142   | 18-620  | 18-628  | 18-636  |        |         |         |        |  |
| NS\$SES | = 000001   | #6-59      |         |         |         |         |        |         |         |        |  |
| NS\$SMC | = ***** GX | 18-684     |         |         |         |         |        |         |         |        |  |
| NS\$VCT | = *****    | 9-232      | 11-370  | 18-660  | 18-682  | 19-717  | 20-799 |         |         |        |  |
| N.CACC  | 000116     | 19-737     |         |         |         |         |        |         |         |        |  |
| N.CIDC  | 000062     | 19-733     | 19-736  | 19-737  |         |         |        |         |         |        |  |
| N.CTL   | 000000     | 19-719     |         |         |         |         |        |         |         |        |  |
| OPN     | 000024 R   | 7-89       | #8-117  |         |         |         |        |         |         |        |  |
| PEM     | 000350 R   | 7-97       | #18-624 |         |         |         |        |         |         |        |  |
| QUETSK  | = ***** GX | 19-769     |         |         |         |         |        |         |         |        |  |
| RDBSZ   | = ***** GX | 18-639     |         |         |         |         |        |         |         |        |  |
| REJECT  | = ***** GX | 8-157      | 19-773  |         |         |         |        |         |         |        |  |
| RETRES  | = ***** GX | 16-578     |         |         |         |         |        |         |         |        |  |
| RMVWND  | = ***** GX | 9-240      |         |         |         |         |        |         |         |        |  |

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE    | REFERENCES                               |
|---------|----------|------------------------------------------|
| AC\$DNT | = 000002 | #6-51                                    |
| AC\$X25 | = 000001 | #6-51                                    |
| AE\$CIR | = 000003 | #6-51                                    |
| AE\$LIN | = 000001 | #6-51                                    |
| AE\$MOD | = 000004 | #6-51                                    |
| CELOG   | = *****  | 9-192                                    |
| CL\$ASZ | = 010500 | #6-51                                    |
| CL\$DLL | = 000500 | #6-51 6-51 6-51 6-51 6-51 6-51 6-51 6-51 |
| CL\$ECL | = 000300 | #6-51 6-51                               |
| CL\$LDN | = 010400 | #6-51 6-51                               |
| CL\$MAN | = 000000 | #6-51 6-51 6-51                          |
| CL\$PAZ | = 034100 | #6-51 6-51 6-51                          |
| CL\$PLH | = 034000 | #6-51 6-51 6-51 6-51                     |
| CL\$PLL | = 000600 | #6-51                                    |
| CL\$PRT | = 034200 | #6-51                                    |
| CL\$ROU | = 010000 | #6-51 6-51 6-51                          |
| CL\$SES | = 000200 | #6-51 6-51 6-51                          |
| CL\$SGE | = 035000 | #6-51 6-51 6-51                          |
| CL\$SSE | = 035100 | #6-51 6-51 6-51 6-51 6-51 6-51 6-51      |
| CL\$TRN | = 000400 | #6-51 6-51 6-51 6-51 6-51 6-51 6-51 6-51 |
| CL\$XL2 | = 013700 | #6-51                                    |
| CL\$XL3 | = 013600 | #6-51 6-51                               |
| CL\$X2S | = 013500 | #6-51 6-51                               |
| CTRSES  | = 000000 | #7-72                                    |
| CTRTBL  | = 000066 | #7-85 #7-97                              |
| DL\$AST | = 000002 | #6-51                                    |
| DL\$HLT | = 000000 | #6-51                                    |
| DL\$IST | = 000001 | #6-51                                    |
| DL\$MAI | = 000004 | #6-51                                    |
| DL\$OFF | = 000001 | #6-51                                    |
| DL\$ON  | = 000000 | #6-51                                    |
| DL\$RUN | = 000003 | #6-51                                    |
| DL\$SHU | = 000002 | #6-51                                    |
| DL\$SYN | = 000005 | #6-51                                    |
| EF\$ACT | = 000001 | #6-51                                    |
| ENCCIR  | = 000134 | R 7-100 #8-148                           |
| ENCSR   | = 000114 | R 7-99 #8-137                            |
| ENC1W   | = 000076 | R 7-97 #8-121                            |
| ENC2W   | = 000106 | R 7-98 #8-130                            |
| EVDSC   | = *****  | GX 9-178                                 |
| EVLSSES | = 000150 | RG #9-176                                |
| EV\$ACF | = 000201 | #6-51                                    |
| EV\$ADR | = 000420 | #6-51                                    |
| EV\$ADU | = 000417 | #6-51                                    |
| EV\$APL | = 000400 | #6-51                                    |
| EV\$ARC | = 000421 | #6-51                                    |
| EV\$AUC | = 000010 | #6-51                                    |
| EV\$AUS | = 000003 | #6-51                                    |
| EV\$CDF | = 000520 | #6-51                                    |



```

128 .SBTTL Executive vector table
129
130 000246 $VECTB::
131 000246 000000 ; Flag for vector not yet filled
132 000250 000000G K SAR6:: .WORD KISAR6 ; RSX exec references
133 000252 000000G TLGTH:: .WORD 7.LGTH
134 000254 000000G USAR6:: .WORD UISAR6
135 000256 000000G ALOCB:: .WORD $ALOCB
136 000260 000000G BLXIO:: .WORD $BLXIO
137 000262 000000G DEACB:: .WORD $DEACB
138 000264 000000G DEVHD:: .WORD $DEVHD
139 000266 000000G DREXT:: .WORD $DREXT
140 000270 000000G FMASK:: .WORD $FMASK
141 000272 000000G IOFIN:: .WORD $IOFIN
142 000274 000000G QRMVF:: .WORD $QRMVF
143 000276 000000G QUEBF:: .WORD $QUEBF
144
145 .IF DF R$$MPL
146 SRPRO:: .WORD $SRPRO
147 .ENDC
148
149 000300 000000G SRSTD:: .WORD $SRSTD
150 000302 000000G STPCT:: .WORD $STPCT
151
152 .IF DF N$$MCP
153 TCBCP:: .WORD $TCBCP
154 .ENDC
155
156 000304 000000G TKTCB:: .WORD $TKTCB
157 000306 000000G TSKRT:: .WORD $TSKRT
158 000310 000000G TTNS:: .WORD $TTNS
159 000312 000000G CALLX:: .WORD $CALLX ; Comm exec references
160 000314 000000G CCBGT:: .WORD $CCBGT
161 000316 000000G CCBRT:: .WORD $CCBRT
162 000320 000000G CEACC:: .WORD $CEACC
163 000322 000000G CELOG:: .WORD $CELOG
164
165 .IF NDF R$$EIS
166 000324 000000G CEDIV:: .WORD $CEDIV
167 000326 000000G CEMUL:: .WORD $CEMUL
168 .ENDC
169
170 000330 000000G CMPDV:: .WORD $CMPDV
171 000332 000000G CSBGT:: .WORD $CSBGT
172 000334 000000G CSBRT:: .WORD $CSBRT
173 000336 000000G DECPY:: .WORD $DECPY
174 000340 000000G EVDSC:: .WORD $EVDSC
175 000342 000000G LDBGY:: .WORD $LDBGY
176 000344 000000G LDBRT:: .WORD $LDBRT
177 000346 000000G LLCRS:: .WORD $LLCRS
178 000350 000000G MVFBF:: .WORD $MVFBF
179 000352 000000G NMCLH:: .WORD $NMCLH
180 000354 000000G OBJHD:: .WORD $OBJHD
181 000356 000000G PDVID:: .WORD $PDVID
182 000360 000000G PDVTA:: .WORD $PDVTA
183 000362 000000G RDBRT:: .WORD $RDBRT
184 000364 000000G RDBSZ:: .WORD $RDBSZ

```

|                   |                  |                |                 |                   |
|-------------------|------------------|----------------|-----------------|-------------------|
| ABT 000016R       | IE.BAD= ***** GX | L.LIA 000034   | M\$MGE= 000000  | P\$SWRD= 000000   |
| ACCLLT= ***** GX  | IN.DAT= 000400   | L.LLA 000002   | M\$MUP= 000000  | Q\$SOPT= 000010   |
| A\$CHK= 000000    | IN.ILS= 000001   | L.LNO 000124   | M\$NET= 000000  | REJ 000076R       |
| A\$CPS= 000000    | IOERR= ***** GX  | L.LNO 000026   | M\$OVR= 000000  | R\$SDER= 000000   |
| A\$PRI= 000000    | IORED= ***** GX  | L.LPT 000065   | M.MAIL 000014   | R\$SK11= 000001   |
| A\$TRP= 000000    | IOSUC= ***** GX  | L.LSA 000030   | M.MAX 000011    | R\$SNSD= 000000   |
| BRKLNK= ***** GX  | IS\$RAR= 000000  | L.LSFD 000046  | M.MBL 000020    | R\$S11M= 000000   |
| CL\$MFL= 000010   | IS\$RDN= 000000  | L.LSFI 000044  | M.NAST 000007   | R\$S11S= 000000   |
| CL\$SFL= 000004   | I.PRM= ***** GX  | L.LTT 000062   | M.NEXT 000002   | SAVOPT= ***** GX  |
| CL\$TYP= 000001   | K\$CNT= 177546   | L.MASQ 000070  | M.RESP 000016   | SNDDIS= ***** GX  |
| CL.MU1= 000001    | K\$CSR= 177546   | L.MAST 000073  | M.SPA 000012    | STCC= 000004      |
| CL.MU2= 000002    | K\$SLDC= 000000  | L.MASZ 000072  | M.TASK 000004   | STCIS= 000006     |
| CL.RES= 177774    | K\$TPS= 000074   | L.NIN 000020   | M.USE 000010    | STCIS= 000002     |
| CM.CON= 000200    | LA.ACK= 100000   | L.NXN 000016   | NC.FM0= 000000  | STSDAT= 000010    |
| CPYOPT= ***** GX  | LA.CRS= 020000   | L.NXTH 000010  | NC.FM1= 000001  | STDIP= 000012     |
| CV\$MSK= 000003   | LA.MSK= 170000   | L.OPD 000103   | NC.FM2= 000002  | STSDIP= 000014    |
| CV\$31= 000001    | LA.NAK= 110000   | L.OPDL 000102  | NF\$BLK= 000100 | S\$SWRG= 000000   |
| CV\$32= 000000    | LA.NMS= 010000   | L.REM 000006   | NF\$DMO= 000010 | S\$SYZ= 007600    |
| CV\$40= 000002    | LA.RES= 040000   | L.RFC 000050   | NF\$MOU= 000040 | TLACHK= ***** GX  |
| CX.GDQ= 000001    | LA.WND= 004000   | L.RLA 000004   | NF\$RST= 000002 | T\$KMG= 000000    |
| CX.REM= 000020    | LD\$LP= 000000   | L.RNO 000022   | NF\$SCN= 000020 | T\$MIN= 000000    |
| CX.REQ= 000002    | LF.DRD= 000004   | L.RTO 000060   | NF\$SHU= 000004 | US\$CNF= 000002   |
| CX.RUI= 000040    | LF.FRC= 000001   | L.RTYD 000055  | NF\$TIM= 000200 | US\$DIS= 000006   |
| CX.SMC= 000010    | LF.HFO= 000010   | L.RTYI 000057  | NS\$DON= 000000 | US\$DON= 000000   |
| CX.UNL= 000004    | LF.HMF= 000040   | L.SEC 000064   | NS\$SDI= 000002 | US\$DSC= 000004   |
| C\$SORE= 000400   | LF.HSF= 000020   | L.SEGZ 000076  | NS\$WDC= 000004 | US\$EAC= 000012   |
| C\$RSH= 177564    | LF.IRD= 000002   | L.LSTA 000000  | NSACQ 000000    | US\$WDS= 000010   |
| DISTBL 000014R    | LF.MMF= 000200   | L.TC 000042    | NSACTL 000032   | V\$CTR= 001000    |
| DSC 000000R       | LF.MSF= 000100   | L.TIC 000043   | NSCIR 000034    | WK.ACK= 000001    |
| D\$RUG= 177514    | LS.DLS= 100000   | L.TIPD 000013  | NSDLA 000020    | WK.AST= 000200    |
| D\$ISK= 000000    | LS.FCC= 000004   | L.TIPI 000012  | NSDLY 000014    | WK.DIS= 000010    |
| D\$SL11= 000001   | LS.FCO= 000001   | L.TMRD 000054  | NSLEN 000054    | WK.INT= 000020    |
| D\$SYNC= 000000   | LS.FCI= 000002   | L.TMRI 000056  | NSENC 000042    | WK.RCV= 000004    |
| D\$YNM= 000000    | LS.ILS= 100000   | L.TYP 000001   | NSERRC 000022   | WK.SND= 000002    |
| ERSABM= 000010    | LS.MAK= 000020   | L.USA 000024   | NSFLG 000005    | WS.DIP= 000010    |
| ERSABO= 000046    | LS.MNK= 000040   | L.USTA 000036  | NSFNC 000006    | WS.INT= 000002    |
| ERSABT= 000011    | LS.RES= 000360   | L.VER 000015   | NSGENQ 000052   | WS.KAS= 000004    |
| ERSACC= 000042    | LS.RSV= 000300   | L.WIND 000040  | NSGTM 000015    | WS.STA= 000001    |
| ERSCDI= 000052    | LT.CCA= 000020   | MA.CI= 000040  | NSHIGH 000033   | W.CINT 000022     |
| ERSCOM= 000047    | LT.DIR= 000010   | MA.DA= 000000  | NSLLT 000026    | W.CSND 000020     |
| ERSFMT= 000005    | LT.LCL= 000001   | MA.II= 000020  | NSLLTM 000024   | W.CTL 000000      |
| ERSMLB= 000006    | LT.LPL= 000002   | MC.CC= 000040  | NSLVC 000036    | W.KAST 000014     |
| ERSNNF= 000012    | LT.NOT= 000040   | MC.CI= 000020  | NSMBXQ 000050   | W.LLT 000004      |
| ERSNOD= 000002    | LT.RSU= 000200   | MC.DC= 000100  | NSPLLT 000030   | W.LUN 000003      |
| ERSNSI= 000013    | LT.SLI= 000004   | MC.DI= 000060  | NSSLA 000016    | W.MBOX 000012     |
| ERSNSR= 000003    | LT.TDA= 000100   | MC.NO= 000000  | NSSNOD 000012   | W.RCVQ 000024     |
| ERSRES= 000001    | L\$ASG= 000000   | MC.RC= 000140  | NSTIM 000004    | W.SEGZ 000006     |
| ERSSTA= 000051    | L\$DRV= 000000   | MD.BM= 000040  | NSVCB 000010    | W.SNDQ 000016     |
| ERSUOB= 000004    | L\$P11= 000001   | MD.EM= 000100  | NS\$ACC= 000001 | W.STAT 000002     |
| E\$XPR= 000000    | L\$11R= 000000   | MD.ILS= 000040 | NS\$EVL= 000001 | W.TMP 000010      |
| FL\$HIO= ***** GX | L.CSTA 000037    | MD.IM= 000020  | NS\$LDV= 000001 | W.WBL 000026      |
| FL\$HMB= ***** GX | L.CTR 000074     | MF.ACK= 000004 | NS\$MLL= 000001 | X\$SDBT= 000000   |
| F\$SLVL= 000001   | L.DCR 000100     | MF.CTL= 000010 | NS\$MOV= 000010 | \$CALLX= ***** GX |
| G\$STYP= 000000   | L.FLAG 000014    | MF.DAT= 000000 | NS\$NCT= 000001 | \$DSOIO 000000RG  |
| G\$STSS= 000000   | L.ILSQ 000052    | M\$CRB= 000124 | NS\$PEM= 000001 | \$MAIBX= ***** GX |
| G\$STTK= 000000   | L.ILTT 000066    | M\$CRX= 000000 | NS\$SES= 000001 | \$REASN= ***** GX |
| G\$SWRD= 000000   | L.LDA 000032     | M\$FCS= 000000 | P\$SP45= 000000 | \$SSHFT= 000001   |

SESDM011S CREATED BY MACRO ON 28-JUN-85 AT 19:55

PAGE 2 H 15

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE    | REFERENCES |
|----------|----------|------------|
| EV\$HCE  | = 035114 | #6-48      |
| EV\$HCI  | = 035113 | #6-48      |
| EV\$HFE  | = 000506 | #6-48      |
| EV\$IFL  | = 000413 | #6-48      |
| EV\$IFO  | = 000415 | #6-48      |
| EV\$IFS  | = 000414 | #6-48      |
| EV\$INF  | = 000515 | #6-48      |
| EV\$LDL  | = 000407 | #6-48      |
| EV\$LDN  | = 010416 | #6-48      |
| EV\$LDO  | = 000411 | #6-48      |
| EV\$LDS  | = 000410 | #6-48      |
| EV\$LSC  | = 000500 | #6-48      |
| EV\$LUP  | = 000412 | #6-48      |
| EV\$NOL  | = 000402 | #6-48      |
| EV\$NRC  | = 000416 | #6-48      |
| EV\$NSC  | = 000200 | #6-48      |
| EV\$NUL  | = 000401 | #6-48      |
| EV\$NVR  | = 000406 | #6-48      |
| EV\$OPL  | = 000403 | #6-48      |
| EV\$PCC  | = 034000 | #6-48      |
| EV\$PCI  | = 034001 | #6-48      |
| EV\$PCM  | = 034002 | #6-48      |
| EV\$PFE  | = 000404 | #6-48      |
| EV\$PPC  | = 034003 | #6-48      |
| EV\$RCF  | = 000517 | #6-48      |
| EV\$RDC  | = 010001 | #6-48      |
| EV\$RDR  | = 010002 | #6-48      |
| EV\$RJE  | = 035106 | #6-48      |
| EV\$RSC  | = 000501 | #6-48      |
| EV\$RUL  | = 000405 | #6-48      |
| EV\$SNA  | = 035000 | #6-48      |
| EV\$SNF  | = 000516 | #6-48      |
| EV\$SPE  | = 035001 | #6-48      |
| EV\$XCE  | = 034110 | #6-48      |
| EV\$XDI  | = 013600 | #6-48      |
| EV\$XGW  | = 034111 | #6-48      |
| EV\$XXM  | = 000514 | #6-48      |
| EV\$XRS  | = 000512 | #6-48      |
| EV\$XSC  | = 000513 | #6-48      |
| EV\$X2S  | = 013500 | #6-48      |
| EV.CCB   | = 000001 | #6-48      |
| EV.CIR   | = 000020 | #6-48      |
| EV.LCB   | = 000100 | #6-48      |
| EV.LIN   | = 000004 | #6-48      |
| EV.MAP   | = 000002 | #6-48      |
| EV.MOD   | = 000040 | #6-48      |
| EV.NOD   | = 000010 | #6-48      |
| EV.PRT   | = 000200 | #6-48      |
| EV\$DATA | = 000020 | #6-48      |
| EV\$EVS  | = 000000 | #6-48      |
| EV\$LCN  | = 000016 | #6-48      |
| EV\$LIN  | = 000000 | #6-48      |

```

223 .SBTTL Scan general delivery queue
224
225 **--SCNGNQ--Scan general delivery queue
226
227 Scan the general delivery queue for tasks which must be requested.
228
229 -
230 Inputs:
231 R5 = Address of database descriptor
232
233 Registers modified:
234 R0, R1, R2, R3, R4
235
236 .PSECT $HIGH
237
238 000034 142765 000020 000005 SCNGNQ: BICB #NFSCN,NFLG(R5)
239 000042 010502 MOV R5,R2 ; Compute address of general delivery
240 000044 062702 000052 ADD #N$GENQ,R2 ; queue listhead
241
242 000050 011204 10$: MOV (R2),R4 ; Get first entry on the queue
243 000052 001443 BEQ 50$; If EQ, queue is empty
244
245 000054 105365 000015 DECB N$GTM(R5) ; Decrement timeout
246 000060 001020 BNE 20$; If NE, timer still active
247 000062 017700 000000G MOV @DECP1,R0 ; Get address of DEC home block
248 000066 116065 000042 000015 MOVB D$INCT(R0),N$GTM(R5) ; Restart the general delivery timer
249 000074 011412 MOV (R4),(R2) ; Unlink CCB from the queue
250 000076 105764 000010 TSTB C,FNC(R4) ; Is it a connect/verification type
251 000102 100005 BPL 15$; If PL, no -its a PEM
252 000104 012701 000046 MOV #ERSABD,R1 ; Set reason for rejection
253 000110 CALL REJECT ; Reject the connection
254 000114 000402 BR 20$
255 000116 15$: CALL @RDBRT ; Just return the buffer
256
257 000122 011204 20$: MOV (R2),R4 ; Get next entry in queue
258 000124 001410 BEQ 40$; If EQ, no more
259 000126 132764 000002 000011 BITB #CX.REQ.C.MOD(R4) ; Does this entry need requesting?
260 000134 001402 BEQ 30$; If EQ, no
261 000136 CALL REQTSK ; Request task to run
262
263 000142 010402 30$: MOV R4,R2 ; Move down queue
264 000144 000766 BR 20$; and keep looking
265
266 000146 005765 000052 40$: TST N$GENQ(R5) ; Anything still in general delivery queue?
267 000152 001403 BEQ 50$; If EQ, no
268 000154 152765 000200 000005 BISB #NFTIM,NFLG(R5); Show we still need timer support
269 000162 50$: RETURN

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54

.TITLE AXMDC - AUX MODEM CONTROLLER  
.IDENT /V05.00/

COPYRIGHT (C) 1978,1979,1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A  
SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR  
ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE  
MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH  
SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE  
TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN  
IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT  
NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL  
EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF  
ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

#### MODULE DESCRIPTION

AUX MODEM CONTROLLER AND AUTODIAL INITIATOR  
DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

#### IDENT HISTORY:

- 1.00 10-FEB-78  
VERSION 2.0 RELEASE
- 2.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

```

438 .SBTTL ST.CDL - CONNECT DELAY
439
440 ***.COND-CONNECT DELAY
441
442 ON ASYNCHRONOUS LINES WITH CARRIER CHECK THAT CARRIER STAYS ASSERTED
443 FOR A MINIMUM LENGTH OF TIME.
444
445 INPUTS:
446 R2 - POINTER TO SYSTEM LINE TABLE
447 R4 - SYSTEM LINE NUMBER
448 R5 - POINTER TO LINE ENTRY IN MDC DATABASE
449
450 .IF NDF X$$D52
451 000554 132765 000001 000004 .COND: BITB #MC.CAR,M.CSV(R5) ; IS CARRIER STILL ASSERTED?
452 000562 001404 ; BEQ 10$; NO ... GO BACK TO LOOKING
453 000564 105765 000002 ; TSTB M.TIM(R5) ; HAS THE TIMER EXPIRED?
454 000570 001407 ; BEQ CNCMP ; YES ... COMPLETE THE CONNECT
455 000572 ; RETURN ; NO ... KEEP WAITING
456 000574 105065 000002 10$: CLRB M.TIM(R5) ; CARRIER HAS BEEN LOST
457 000600 112765 000004 000003 20$: MOVB #ST.WCN,M.STT(R5) ; RETURN TO WAIT FOR CONNECT STATE
458 000606 ; RETURN
459
460 .IFF
461
462 ; When interfacing with a modem which generates continuous carrier,
463 ; wait until both Carrier Detect and Clear To Send are asserted.
464
465 .COND: TSTB M.TIM(R5) ; If timer has expired
466 ; BEQ 5$; then
467 ; RETURN
468 5$: MOVB #<TM.CON-TM.CHK>, M.TIM(R5) ; Start non-data call timer
469 ; MOVB #ST.DLY, M.STT(R5)
470
471 ; Carrier and Clear to Send must become asserted with in a certain
472 ; amount of time, else a disconnect is posted.
473
474 .COND2: BITB #MC.DSR, M.CSV(R5) ; If Data Set Ready still asserted
475 ; BEQ 20$; then
476 ; BITB #MC.CAR, M.CSV(R5) ; If Carrier is asserted
477 ; BEQ 10$; and
478 ; BITB #MC.CTS, M.CSV(R5) ; If Clear To Send is asserted
479 ; BEQ 10$; then
480 ; CLRB M.TIM(R5) ; Stop connect timer and
481 ; BR CNCMP ; Goto complete the connect
482 10$: ; else
483 ; TSTB M.TIM(R5) ; If timer has expired
484 ; BEQ 25$; then Goto post disconnect
485 ; RETURN ; else Return and wait
486 20$: ; else (DSR no longer asserted)
487 ; CLRB M.TIM(R5) ; Abort the non-data call timer
488 ; MOVB #ST.CFL, M.STT(R5) ; Set state to Connect Failure
489 ; ; Fall thru to post disconnect
490
491 ; The connect attempt has failed. Post an Enable Complete followed by an
492 ; immediate Asynchronous Disconnect Detect.
493
494 .CONFL: CALL CNCMP ; Post an Enable Complete

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE    | REFERENCES                                                           |
|---------|----------|----------------------------------------------------------------------|
| L.CTL   | 000012   | #5-66                                                                |
| L.CVA   | 177776   | #5-66                                                                |
| L.DDM   | 000002   | #5-66                                                                |
| L.DDS   | 000004   | #5-66                                                                |
| L.DLC   | 000003   | #5-66                                                                |
| L.DLM   | 000006   | #5-66                                                                |
| L.DLS   | 000010   | #5-66                                                                |
| L.FLG   | 000000   | #5-66                                                                |
| L.KRBA  | 000016   | #5-66                                                                |
| L.LEN   | = 000022 | #5-66                                                                |
| L.MPF   | 000022   | #5-66                                                                |
| L.NMST  | 000020   | #5-66                                                                |
| L.NSTA  | 000014   | #5-66                                                                |
| L.OWNR  | 000021   | #5-66                                                                |
| L.UNT   | 000013   | #5-66                                                                |
| MC.CAR  | = 000001 | 15-415 16-451 18-540 19-580 21-625 22-645 23-669                     |
| MC.CCB  | = 000000 | 10-255 13-373 17-518 20-606                                          |
| MC.DSR  | = 000002 | 15-420 18-545                                                        |
| MC.RNG  | = 000004 | 14-390                                                               |
| MDMCTL  | 000022   | RG #7-148 7-153                                                      |
| MDMSCN  | 000310   | RG #13-323 13-375                                                    |
| MDMSRV  | 000000   | R #6-114 13-371                                                      |
| MS.SCA  | = 000002 | 13-327                                                               |
| MS.SYN  | = 000001 | 15-413 18-538                                                        |
| M.CSV   | 000004   | *8-180 *9-201 *12-306 *13-362 13-367 13-372 *13-373 14-390 15-415    |
|         |          | 15-420 16-451 *17-518 18-540 18-545 19-580 *20-606 *22-645 *23-669   |
| M.LEN   | 000010   | 7-152 12-304 13-374                                                  |
| M.LIN   | 000001   | 7-150 12-302 13-329 13-369 24-696                                    |
| M.PSV   | 000005   | *8-179 *9-200 *10-255 13-367 *13-372 21-625                          |
| M.STA   | 000006   | *7-155 24-697                                                        |
| M.STT   | 000003   | 8-176 *8-178 *9-199 10-237 10-239 *10-245 *10-254 13-325 13-370      |
|         |          | *15-418 *16-457 *18-543 *19-586 *20-602 *24-694                      |
| M.TIM   | 000002   | *10-236 13-363 *13-365 *15-417 16-453 *16-456 *18-542 19-582 *19-585 |
| N\$1LN  | = *****  | 23-662                                                               |
| PDVTA   | = *****  | 13-331                                                               |
| RING    | 000066   | R 12-297                                                             |
| R\$1MPL | = *****  | 7-161 #8-176                                                         |
| R\$1PRO | = *****  | 13-348                                                               |
| R\$11D  | = *****  | 5-59 9-202                                                           |
| R\$11M  | = 000000 | 5-63 12-283                                                          |
| R\$11S  | = *****  | 5-63                                                                 |
| SF.ACT  | = 000200 | #5-66                                                                |
| SF.ENA  | = 000100 | #5-66                                                                |
| SF.LPB  | = 000004 | #5-66                                                                |
| SF.MFL  | = 000040 | #5-66                                                                |
| SF.PAC  | = 000020 | #5-66                                                                |
| SF.REA  | = 000010 | #5-66                                                                |
| SF.SER  | = 000001 | #5-66                                                                |
| SF.SVC  | = 000002 | #5-66                                                                |
| SF.UNL  | = 000040 | #5-66                                                                |
| SL.TMA  | = *****  | GX 1-140                                                             |

AXSCH      CREATED BY    MACRO    ON 28-JUN-85 AT 18:30      PAGE 1      I 4

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE    | REFERENCES                                                  |
|--------|----------|-------------------------------------------------------------|
| CF.DDM | = 000002 | #5-61                                                       |
| CF.DYN | = 000004 | #5-61                                                       |
| CF.EIS | = 000010 | #5-61                                                       |
| CF.FRK | = 100000 | #5-61      6-96      6-215                                  |
| CF.LOG | = 000020 | #5-61                                                       |
| CF.MDM | = 000001 | #5-61                                                       |
| CF.TIM | = 000400 | #5-61                                                       |
| CMFRK  | = *****  | GX      6-98      6-194                                     |
| CS.LST | = 040000 | 6-138      6-158                                            |
| CXOPT  | = *****  | GX      6-96      6-215                                     |
| C.BID  | 000003   | 6-166      *6-170                                           |
| C.FNC  | 000010   | 6-173                                                       |
| C.LIN  | 000006   | 6-159      *6-168                                           |
| C.STS  | 000012   | 6-138      *6-158                                           |
| INTCT  | = *****  | GX      6-112                                               |
| IS\$AS | = *****  | 5-62                                                        |
| PDDSP  | = *****  | GX      6-178                                               |
| PDSP   | = *****  | GX      6-186                                               |
| PDVTA  | = *****  | GX      6-175                                               |
| PR7    | = *****  | GX      6-100                                               |
| PS     | = *****  | GX      *6-100      *6-156      *6-187      *6-227          |
| R\$MPL | = *****  | 5-64      6-102      6-118      6-146      6-197      6-217 |
| R\$11D | = *****  | 5-62                                                        |
| R\$11M | = 000000 | 5-62                                                        |
| R\$11S | = *****  | 5-62                                                        |
| STDLC  | = *****  | GX      6-185                                               |
| TKPS   | = *****  | GX      6-112                                               |
| X\$MCB | = *****  | 5-62      5-62                                              |
| ZF.COU | = 001000 | #5-62                                                       |
| ZF.DDM | = 000001 | #5-62                                                       |
| ZF.DIA | = 004000 | #5-62                                                       |
| ZF.DLC | = 000002 | #5-62                                                       |
| ZF.DVP | = 100000 | #5-62                                                       |
| ZF.INI | = 040000 | #5-62                                                       |
| ZF.KMX | = 000020 | #5-62                                                       |
| ZF.LLC | = 000004 | #5-62                                                       |
| ZF.LMC | = 000100 | #5-62                                                       |
| ZF.MAN | = 020000 | #5-62                                                       |
| ZF.MFL | = 000010 | #5-62                                                       |
| ZF.MTM | = 000400 | #5-62                                                       |
| ZF.MUX | = 000040 | #5-62                                                       |
| ZF.PSE | = 002000 | #5-62                                                       |
| ZF.SLI | = 010000 | #5-62                                                       |
| ZF.TIM | = 000200 | #5-62                                                       |
| ZF.X3P | = 000000 | #5-62                                                       |
| ZS.ASN | = 100000 | #5-62                                                       |
| ZS.BSY | = 140000 | #5-62                                                       |
| Z.AVL  | 000014   | #5-62                                                       |
| Z.DAT  | 000016   | #5-62      6-177                                            |
| Z.DSP  | 000000   | #5-62      5-62                                             |
| Z.FLG  | 000010   | #5-62                                                       |
| Z.LEN  | = 000016 | #5-62                                                       |



```

342 .SBTTL $XLINK - Link/unlink block in extended pool
343
344 *--$XLINK-Link/unlink block in extended pool
345
346 This routine is called to link or unlink a block in a list that may
347 include allocations from extended pool.
348
349 Inputs:
350 to link block into list:
351 MOV X,-(SP) ; address of block to be linked into list
352 MOV Y,-(SP) ; address of previous block in list
353 CALLX $XLINK,AUX
354 to unlink block from list:
355 MOV X,-(SP) ; address of previous block in list
356 MOV Y,-(SP) ; address of block to be unlinked from list
357 CALLX $XLINK,AUX
358
359 Outputs:
360 Block is linked/unlinked, addresses are removed from the stack.
361
362 Registers:
363 No registers are modified
364
365 $XLINK::
366 000432
367 000432 010046 MOV R0,-(SP) ; save R0
368
369 .IF DF M$MGE
370
371 000434 032777 000000G 000000G BIT #F2.DAS,@FMSK2 ; is data space enabled?
372 000442 001403 BEQ 10$; br if no
373 000444 016646 000014 MOV 14(SP),-(SP) ; set block address for $CEACC
374 000450 000402 BR 20$
375
376 000452 016646 000012 10$: MOV 12(SP),-(SP) ; set block address for $CEACC
377
378
379 20$: CALL @CEACC ; map to block
380 000456 012600 MOV (SP)+,R0 ; get mapped address
381 000462 011046 MOV (R0),-(SP) ; save address of next block in list
382
383
384 000466 032777 000000G 000000G BIT #F2.DAS,@FMSK2 ; is data space enabled?
385 000474 001403 BEQ 30$; br if no
386 000476 016610 000020 MOV 20(SP),(R0) ; set new link word into block
387 000502 000402 BR 40$
388
389 000504 016610 000016 30$: MOV 16(SP),(R0) ; set new link word into block
390
391 40$: MOV (R0),-(SP) ; copy address of block for $CEACC
392 000510 011046 CALL @CEACC ; map to block
393 000512 012600 MOV (SP)+,R0 ; get mapped address
394 000516 012610 MOV (SP)+,(R0) ; set new link word
395 000522 012600 MOV (SP)+,R0 ; restore R0
396
397 000524 032777 000000G 000000G BIT #F2.DAS,@FMSK2 ; data space enabled?
398 000532 001403 BEQ 50$; br if no

```

```

50 .SBTTL Macro definitions
51
52 ;
53 ; MACRO LIBRARY CALLS
54 ;
55
56 .MCALL INHIB$,ENABL$,SAVRG,RESRG
57 .MCALL SLTDF$,PDVDF$,CLKDF$,STMDF$
58
59 000000 SLTDF$; DEFINE SLT OFFSETS
60 000000 PDVDF$; DEFINE PDV OFFSETS
61 000000 CLKDF$; DEFINE CLOCK BLOCK OFFSETS
62 000000 STMDF$; DEFINE SHORT TIMER OFFSETS

```

AXTIM      CREATED BY MACRO ON 28-JUN-85 AT 18:31      PAGE 1      I 7

SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL  | VALUE      | REFERENCES                  |
|---------|------------|-----------------------------|
| CLCINS  | 000042 R   | #6-93      9-299            |
| CLINS   | = ***** GX | 6-101                       |
| C.SYST  | = 000006   | 6-98                        |
| DSPTM   | = ***** GX | 8-258                       |
| ISSAS   | = *****    | 5-60                        |
| LF.ACT  | = 100000   | #5-59      8-192            |
| LF.BRO  | = 000400   | #5-59                       |
| LF.BWT  | = 000007   | #5-59                       |
| LF.ENA  | = 002000   | #5-59                       |
| LF.LPB  | = 001000   | #5-59                       |
| LF.MDC  | = 000100   | #5-59                       |
| LF.MFL  | = 004000   | #5-59                       |
| LF.MTP  | = 000020   | #5-59                       |
| LF.PAC  | = 000200   | #5-59                       |
| LF.RDY  | = 040000   | #5-59                       |
| LF.REA  | = 010000   | #5-59                       |
| LF.SER  | = 000040   | #5-59                       |
| LF.TIM  | = 000010   | #5-59      8-194            |
| LF.UNL  | = 020000   | #5-59                       |
| LF.X2P  | = 000000   | #5-59                       |
| LN.CLO  | = 000000   | #5-59                       |
| LN.DUM  | = 000005   | #5-59                       |
| LN.LOA  | = 000004   | #5-59                       |
| LN.LOO  | = 000003   | #5-59                       |
| LN.OAU  | = 000003   | #5-59                       |
| LN.OFF  | = 000001   | #5-59                       |
| LN.ON   | = 000000   | #5-59                       |
| LN.OOP  | = 000004   | #5-59                       |
| LN.OPE  | = 000001   | #5-59                       |
| LN.REF  | = 000002   | #5-59                       |
| LN.SER  | = 000002   | #5-59                       |
| LN.STA  | = 000017   | #5-59                       |
| LN.SUB  | = 000360   | #5-59                       |
| LN.TRI  | = 000006   | #5-59                       |
| LSS11   | = *****    | 12-430      12-455          |
| L.COST  | 000015     | #5-59                       |
| L.CTL   | 000012     | #5-59                       |
| L.CVA   | 177776     | #5-59                       |
| L.DDM   | 000002     | #5-59      8-211      8-215 |
| L.DDS   | 000004     | #5-59      8-210            |
| L.DLC   | 000003     | #5-59      8-215      8-218 |
| L.DLM   | 000006     | #5-59                       |
| L.DLS   | 000010     | #5-59      8-217            |
| L.FLG   | 000000     | #5-59                       |
| L.KRBA  | 000016     | #5-59                       |
| L.LEN   | = 000022   | #5-59                       |
| L.MPF   | 000022     | #5-59                       |
| L.NMST  | 000020     | #5-59                       |
| L.NSTA  | 000014     | #5-59                       |
| L.OWNR  | 000021     | #5-59                       |
| L.UNT   | 000013     | #5-59                       |
| N\$6TLN | = *****    | 8-221                       |

```

261
262
263
264
265
266
267
268 000536 012764 000201 000010
269
270
271
272
273
274
275
276
277
278 000544 132701 000100
279 000550 001403
280 000552 152764 000040 000011
281
282 000560
283
284 000560 016700 000000G
285 000564 016000 000000G
286 000570 032700 000001
287 000574 001423
288 000576 032700 100000
289 000602 001420
290
291 000604 042701 177774
292
293
294 000610 020127 000002
295 000614 001413
296
297
298 000616 006301
299 000620 016564 000012 000030
300 000626 110164 000033
301 000632 112764 000207 000010
302 000640 012703 000000G
303
304
305
306
307 000644
308
309
310 000644
311 000650 103020
312
313
314
315 000652 122764 000201 000010
316 000660 001413
317

```

```

 .IF DF N$$MCP
MOV B $RQCPY,C.STS(R4); And the max # of copies allowed
 .ENDC
MOV #NT.CON!CM.CON,C.FNC(R4)
 .IF DF N$$MCP
BIT B #OF.SMC,R1 ; Spawn multiple copies of this task?
BEQ 30$; If EQ, no
BIS B #CX.SMC,C.MOD(R4)
30$:
 .ENDC
BIT B #OF.RLU,R1 ; Run task under login UIC?
BEQ 40$; If EQ, no
BIS B #CX.RUI,C.MOD(R4)
40$:
 .IF DF N$$ACC
MOV $UCB,R0 ; Get address of UCB
MOV U.CW3(R0),R0 ; Get device features mask
BIT #NF.ACC,R0 ; Is access control supported in this system?
BEQ 50$; If EQ, no
BIT #NF$ACC,R0 ; Is access control enabled?
BEQ 50$; If EQ, no
BIC #^C<3>,R1 ; Isolate verification level
 .IF NDF R$$PRO ; Always call login for P/OS
R1,#2 ; Is verification required?
BEQ 50$; If EQ, no
 .ENDC ; NDF R$$PRO
ASL R1 ; Form word index
MOV N$$SNOD(R5),C.CNT2(R4); Pass node number to NVP
MOV R1,C.FLG2+1(R4); and save for later
MOV #NT.VFY!CM.CON,C.FNC(R4)
MOV #VFYNAM,R3 ; Point to verification task name
 .IF DF R$$PRO
MOV $COPT,C.PRO(R4); Run NVP under CO:
 .ENDC ; DF R$$PRO
50$:
 .ENDC
CALL QUETSK ; Queue request to task
BCC 110$; If CC, request successfully queued
 .IF DF N$$ACC
CMPB #NT.CON!CM.CON,C.FNC(R4)
BEQ 100$; If EQ, the connect request failed

```

```

42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,CALLE,SAVMAP,RESMAP,RECMAP,MAPLLT
45 .MCALL $IERRC
46 .MCALL CCBDF$,ECDDb$,MBXDF$,LLWDF$,CNBDF$,NSSYM$,MSGDF$
47 .MCALL LLTDF$,DHBDF$
48
49 CCBDF$; Define CCB offsets
50 ECDDb$; Define ECL database offsets
51 MBXDF$; Define mailbox offsets
52 LLWDF$; Define window block offsets
53 CNBDF$; Define connect pending block offsets
54 NSSYM$; Define NS: symbols
55 MSGDF$; Define message flags
56 LLTDF$; Define LLT offsets
57 DHBDF$; Define DEC home block offsets
58
59 000001 NSSSES = 1 ; This module is part of session control

```

```

515 .SBTTL User/network abort and disconnect data processing
516
517 ;+
518 ;**AB0-Network disconnect
519 ;**AB1-User abort
520 ;**DSC-User disconnect
521
522 ; Process the disconnect/abort event.
523
524 ; Inputs:
525 ; R1 = Address of CCB
526 ; R3 = Address of I/O packet
527 ; R4 = Virtual address of user's buffer
528 ; User's buffer is mapped
529 ; R5 = Address of database descriptor
530 ; 2(SP) - # of bytes to be transferred
531 ; 4(SP) - High byte: Data type code
532 ; Low byte: I/O status return code
533 ; 6(SP) - Address of I/O packet
534
535 ; Registers modified:
536 ; R0, R2, R4
537
538 .ENABL LSB
539 000540 116166 000004 000002 AB0: MOVB C.NSP(R1),2(SP) ; Fill in disconnect error code
540 000546 000405 BR 10$; Enter common code
541
542 000550 ABT:
543 000550 016600 000002 DSC: MOV 2(SP),R0 ; Get # of bytes to transfer
544 000554 001402 BEQ 10$; If EQ, none
545 000556 CALL MOVDAT ; Copy optional data to user buffer
546
547 000562 010104 10$: MOV R1,R4 ; Change register conventions
548 000564 116466 000011 000003 MOVB C.MOD(R4),3(SP) ; Put LUN # in high byte of second I/O status word
549 000572 CALLR DISCMP ; Complete the disconnect
550
551 .DSABL LSB

```

SESC111S CREATED BY MACRO ON 28-JUN-85 AT 19:54 PAGE 3 I 11  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE       | REFERENCES                           |
|---------|-------------|--------------------------------------|
| R\$MPL  | = *****     | 18-674                               |
| R\$PRO  | = *****     | 18-670 19-747                        |
| SPA     | = 000172 R  | 7-91 #10-312                         |
| SRSTD   | = ***** GX  | 18-678                               |
| TDISP   | = 000430 R  | 11-373 #11-385                       |
| TLCHK   | = ***** GX  | 19-722                               |
| T.NAM   | = ***** GX  | 18-625 18-626                        |
| UISAR6  | = ***** GX  | 18-682                               |
| US\$DON | = 000000    | 9-289 9-290 9-292 9-293              |
| US\$DSC | = 000004    | 9-294                                |
| VFY     | = 000446 R  | 11-391 #13-474                       |
| W.LLT   | = 000004    | 9-223                                |
| W.TMP   | = 000010    | 9-226                                |
| X\$SHDR | = *****     | 9-199 9-208 9-271                    |
| \$CALLX | = ***** GX  | 9-232                                |
| \$CLQIO | = 000000 RG | #9-186                               |
| \$CTQIO | = 000000 RG | #7-83                                |
| \$IOPKT | = ***** GX  | *9-226 *9-268 19-726                 |
| \$MAIBX | = ***** GX  | *19-721                              |
| \$RQNAM | = ***** GX  | *18-625 *18-626 18-630 18-667 18-668 |
| \$UCB   | = ***** GX  | 9-214                                |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE    | REFERENCES |
|----------|----------|------------|
| EV\$COZ  | = 000011 | #6-51      |
| EV\$DBR  | = 000302 | #6-51      |
| EV\$GAS  | = 035101 | #6-51      |
| EV\$HCE  | = 035114 | #6-51      |
| EV\$HCI  | = 035113 | #6-51      |
| EV\$HFE  | = 000506 | #6-51      |
| EV\$IFL  | = 000413 | #6-51      |
| EV\$IFO  | = 000415 | #6-51      |
| EV\$IFS  | = 000414 | #6-51      |
| EV\$INF  | = 000515 | #6-51      |
| EV\$LDL  | = 000407 | #6-51      |
| EV\$LDN  | = 010416 | #6-51      |
| EV\$LDO  | = 000411 | #6-51      |
| EV\$LDS  | = 000410 | #6-51      |
| EV\$LSC  | = 000500 | #6-51      |
| EV\$LUP  | = 000412 | #6-51      |
| EV\$NOL  | = 000402 | #6-51      |
| EV\$NRC  | = 000416 | #6-51      |
| EV\$NSC  | = 000200 | #6-51      |
| EV\$NUL  | = 000401 | #6-51      |
| EV\$NVR  | = 000406 | #6-51      |
| EV\$OPL  | = 000403 | #6-51      |
| EV\$PCC  | = 034000 | #6-51      |
| EV\$PCI  | = 034001 | #6-51      |
| EV\$PCM  | = 034002 | #6-51      |
| EV\$PFE  | = 000404 | #6-51      |
| EV\$PPC  | = 034003 | #6-51      |
| EV\$RCF  | = 000517 | #6-51      |
| EV\$RDC  | = 010001 | #6-51      |
| EV\$RDR  | = 010002 | #6-51      |
| EV\$RJE  | = 035106 | #6-51      |
| EV\$RSC  | = 000501 | #6-51      |
| EV\$RUL  | = 000405 | #6-51      |
| EV\$SNA  | = 035000 | #6-51      |
| EV\$SNF  | = 000516 | #6-51      |
| EV\$SPE  | = 035001 | #6-51      |
| EV\$XCE  | = 034110 | #6-51      |
| EV\$XDI  | = 013600 | #6-51      |
| EV\$XGW  | = 034111 | #6-51      |
| EV\$XMX  | = 000514 | #6-51      |
| EV\$XRS  | = 000512 | #6-51      |
| EV\$XSC  | = 000513 | #6-51      |
| EV\$X2S  | = 013500 | #6-51      |
| EV.CCB   | = 000001 | #6-51      |
| EV.CIR   | = 000020 | #6-51      |
| EV.LCB   | = 000100 | #6-51      |
| EV.LIN   | = 000004 | #6-51      |
| EV.MAP   | = 000002 | #6-51      |
| EV.MOD   | = 000040 | #6-51      |
| EV.NOD   | = 000010 | #6-51      |
| EV.PRT   | = 000200 | #6-51      |
| EV\$DATA | = 000020 | #6-51      |



SES DAT - Session control local MACRO V05.03b Friday 28-Jun-85 19:54 <sup>13</sup> Page 9-1  
Executive vector table

|     |        |         |                         |
|-----|--------|---------|-------------------------|
| 185 |        |         | .IF DF R\$\$PRO         |
| 186 |        |         | ZTIM2:: .WORD \$ZTIME+2 |
| 187 |        |         | .IFF : DF R\$\$PRO      |
| 188 | 000366 | 000000G | ZTIM2:: .WORD \$ZTIM2   |
| 189 |        |         | .ENDC : DF R\$\$PRO     |
| 190 |        | 000051  | \$VECLN == .-\$VECTB/2  |
| 191 |        |         |                         |
| 192 |        | 000001  | .END                    |

SESDIS - Session control discon MACRO V05.03b Friday 28-Jun-85 19:55 <sup>1 14</sup> Page 8-2  
Symbol table

. ABS. 000124 000 (RW,I,GBL,ABS,OVR)  
000146 001 (RW,I,LCL,REL,CON)  
\$HIGH 000022 002 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 15488 Words ( 61 Pages)  
Size of core pool: 16552 Words ( 63 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:13.40

SY:SESDIS11S.V2,[131,134]SESDIS11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCS/PA:1,[131,10]V2,SESDIS

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE    | REFERENCES   |
|---------|----------|--------------|
| ESMOD   | 000012   | #6-48        |
| ESNOD   | 000010   | #6-48        |
| ESPORT  | 000014   | #6-48        |
| ESPRM   | 000002   | #6-48        |
| ESSTAT  | 000006   | #6-48        |
| ESTCB   | 000004   | #6-48        |
| CTL     | 000020   | #6-48        |
| DATA    | 000046   | #6-48        |
| EVT     | 000002   | #6-48        |
| LCN     | 000042   | #6-48        |
| LEN     | 000216   | #6-48        |
| LIN     | 000024   | #6-48        |
| LNK     | 000000   | #6-48        |
| MOD     | 000036   | #6-48        |
| NOD     | 000034   | #6-48        |
| PDV     | 000021   | #6-48        |
| PORT    | 000040   | #6-48        |
| PRM     | 000026   | #6-48        |
| PVC     | 000044   | #6-48        |
| SIZ     | 000022   | #6-48        |
| TIME    | 000004   | #6-48        |
| LSLNK   | 000000 R | 7-104 #8-120 |
| FR\$BCC | = 000007 | #6-48        |
| FR\$CCF | = 000001 | #6-48        |
| FR\$CDF | = 000002 | #6-48        |
| FR\$DAO | = 000011 | #6-48        |
| FR\$EXC | = 000000 | #6-48        |
| FR\$FRM | = 000010 | #6-48        |
| FR\$FTL | = 000005 | #6-48        |
| FR\$OPN | = 000004 | #6-48        |
| FR\$RFD | = 000006 | #6-48        |
| FR\$SBU | = 000012 | #6-48        |
| FR\$SHO | = 000003 | #6-48        |
| FR\$UBU | = 000013 | #6-48        |
| FR\$UPT | = 000014 | #6-48        |
| IOSUC   | = *****  | GX 7-91      |
| I.FCN   | = *****  | GX 7-70      |
| KILLNK  | = *****  | GX 8-131     |
| L.DCR   | = 000100 | *8-129       |
| MO\$SAC | = 000016 | #6-48        |
| MO\$SPR | = 000012 | #6-48        |
| MO\$SSV | = 000014 | #6-48        |
| MO\$25A | = 000006 | #6-48        |
| MO\$25P | = 000002 | #6-48        |
| MO\$25S | = 000004 | #6-48        |
| MO\$29S | = 000010 | #6-48        |
| M\$HIGH | = 000003 | #6-48        |
| M\$3100 | = 000000 | #6-48        |
| M\$3101 | = 000001 | #6-48        |
| M\$3102 | = 000002 | #6-48        |
| M\$3103 | = 000003 | 6-48         |
| NF\$DMO | = 000010 | 7-68         |

```

271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286 000164
287
288 000164 016704 000000G AUXTSK: MOV NMCLH,R4 ; Get the first element in the list
289 000170 010402 10$: MOV R4,R2 ; Save the previous link word address
290 000172 011404 MOV (R4),R4 ; Get next element
291 000174 001465 BEQ 50$; If EQ, end of list
292 000176 005764 000034 TST C.ADD(R4) ; Is this for NETACP ?
293 000202 001053 BNE 40$; If NE, no
294 000204 010403 MOV R4,R3 ; Copy the CCB address
295 000206 062703 000036 ADD #C.ADD+2,R3 ; Set up the task name pointer for REQTS1
296
297
298
299
300 000212 012713 131574
301
302
303 000216 012743 046166 MOV #*RLIN,-(R3) ;
304 000222 152764 000010 000011 BISB #CX.SMC,C.MOD(R4) ; Always treat line watcher as multi copy
305 000230 012767 000000G 000000G MOV #N$SMC,$RQCPY ; Set up maximum number of copies
306 000236 016764 000000G 000042 MOV $COPT,C.PRO(R4) ; Run LIN$$$ under CO:
307 000244 CALL REQTS1 ; Request the task
308 000250 103030 BCC 40$; If CC, continue
309
310 000252 011412 15$: MOV (R4),(R2) ; Else, remove this CCB or RDB from the list
311 000254 001005 BNE 25$; If NE, its not at the end of the list
312 000256 016746 000000G MOV NMCLH,-(SP) ; Else, update the tail pointer
313 000262 062716 000002 ADD #2,(SP) ; ...
314 000266 010236 MOV R2,@(SP)+ ; ...
315 000270 016446 000012 000003 25$: MOV C.STS(R4),-(SP) ; Save the chained buffer flag
316 000274 122764 000002 000003 CMPB #CB.CCB,C.BID(R4) ; Return the CCB or RDB to the free list
317 000302 001003 BNE 30$; If NE, its an RDB
318 000304 CALL @CCBRT ; Return the CCB
319 000310 000402 BR 35$; Continue searching the list
320 000312 30$: CALL @RDBRT ; Return the RDB
321 000316 010204 35$: MOV R2,R4 ; Copy the 'previous' element
322 000320 032726 040000 BIT #CS.LST,(SP)+ ; Was this the last one in the chain ?
323 000324 001321 BNE 10$; If NE, yes
324 000326 011404 MOV (R4),R4 ; Get next element in chain
325 000330 000750 BR 15$; Continue returning this chain
326
327 000332 032764 040000 000012 40$: BIT #CS.LST,C.STS(R4) ; Is this the last one ?

```

```

56 .SBTTL Macro definitions
57
58 .MCALL PDVDF$,CCBDF$,MDCDF$,CALLR,SLTDF$
59 .IF DF R$$$PRO
60 .MCALL LINDF$
61 LINDF$; DEFINE PRO/DECnet LINE DESCRIPTOR
62 .ENDC
63 PDVDF$; DEFINE PDV OFFSETS
64 CCBDF$; DEFINE CCB OFFSETS
65 MDCDF$; DEFINE MODEM CONTROLLER CONSTANTS
66 SLTDF$; DEFINE SYSTEM LINE TABLE OFFSETS
67
68 ;
69 ; DEFINE LOCAL STATE VALUES
70 .ASECT
71 000004 000000
72 ST.IDL::BLKW ; LINE IDLE
73 ST.WRG::BLKW ; WAITING FOR RING
74 ST.WCN::BLKW ; WAITING FOR CONNECT
75 ST.CDL::BLKW ; CARRIER DELAY TO SEE IF CONNECTED
76 .IF DF X$$$D52
77 ST.DLY::BLKW ; Delay for Carrier and Clear to Send
78 ST.CFL::BLKW ; Connect failure
79 .ENDC
80 ST.ACT::BLKW ; LINE IS ACTIVE
81 ST.ADL::BLKW ; CARRIER DELAY TO SEE IF LINE IS ACTIVE
82 ST.CER::BLKW ; LOST CARRIER OR DSR WHILE ACTIVE
83 ST.DDL::BLKW ; DISCONNECT DELAY TO ALLOW CIRCUITS TO RESET
84 ST.ABO::BLKW ; POST ABORT TO DLC LEVEL
85
86 000000
87 .PSECT
88 ;
89 ; TIMER VALUES
90 .IF NDF X$$$D52
91 TM.DIS=2 ; TIMER FOR DISCONNECT DELAY
92 TM.CON=2 ; TIMER FOR CARRIER CONNECT DELAY
93 TM.ACT=2 ; TIMER FOR CARRIER ACTIVE DELAY
94 ;
95 .IFF
96 ; All timer values are N + 1 since a request for 1 second timer
97 ; service results in a timer dispatch occurring in 0 to 1 second.
98 ;
99 TM.CHK = 1 + 1 ; Timer for delay before checking CTS & CD
100 TM.CON = 30 + 1 ; Timer to protect against non-data calls
101 TM.ACT = 2 + 1 ; Timer for lost Carrier delay
102 TM.DIS = 5 + 1 ; Timer for disconnect delay
103 .ENDC
104 ;

```

|     |       |        |               |                                 |
|-----|-------|--------|---------------|---------------------------------|
| 495 |       | BCC    | 30\$          | ; CCB allocation successful?    |
| 496 |       | RETURN |               | ; No ... Try again next timeout |
| 497 | 30\$: | MOVB   | M.LIN(R5), R4 | ; Yes ... Pass line number      |
| 498 |       | CALLR  | DISC          | ; Goto post disconnect          |
| 499 |       | .ENDC  |               |                                 |
| 500 | ;     |        |               |                                 |

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE    | REFERENCES |         |        |        |         |        |
|---------|----------|------------|---------|--------|--------|---------|--------|
| ST.ABO  | 000020   | G #5-84    | 10-254  |        |        |         |        |
| ST.ACT  | 000010   | G #5-80    | 17-514  |        |        |         |        |
| ST.ADL  | 000012   | G #5-81    | 18-543  |        |        |         |        |
| ST.CDL  | 000006   | G #5-75    | 10-239  |        |        |         |        |
| ST.CER  | 000014   | G #5-82    | 20-602  |        |        |         |        |
| ST.DDL  | 000016   | G #5-83    | 10-245  |        |        |         |        |
| ST.IDL  | 000000   | G #5-72    | 14-393  |        |        |         |        |
| ST.WCN  | 000004   | G #5-74    | 9-199   |        |        |         |        |
| ST.WRG  | 000002   | G #5-73    | 8-178   |        |        |         |        |
| S.COST  | 000001   | #5-66      |         |        |        |         |        |
| S.FLG   | 000000   | #5-66      |         |        |        |         |        |
| S.LEN   | 000004   | #5-66      |         |        |        |         |        |
| S.NMST  | 000002   | #5-66      |         |        |        |         |        |
| S.OWNR  | 000003   | #5-66      |         |        |        |         |        |
| TM.ACT  | = 000002 | #5-93      | 18-542  |        |        |         |        |
| TM.CON  | = 000002 | #5-92      | 15-417  |        |        |         |        |
| TM.DIS  | = 000002 | #5-91      | 10-236  |        |        |         |        |
| XS\$D52 | = *****  | 5-76       |         | 6-118  | 10-241 | 15-412  | 16-450 |
| XS\$MCB | = *****  | 5-63       |         |        |        |         | 18-537 |
| ZF.COU  | = 001000 | #5-63      |         |        |        |         |        |
| ZF.DDM  | = 000001 | #5-63      |         |        |        |         |        |
| ZF.DIA  | = 004000 | #5-63      |         |        |        |         |        |
| ZF.DLC  | = 000002 | #5-63      |         |        |        |         |        |
| ZF.DVP  | = 100000 | #5-63      |         |        |        |         |        |
| ZF.INI  | = 040000 | #5-63      |         |        |        |         |        |
| ZF.KMX  | = 000020 | #5-63      |         |        |        |         |        |
| ZF.LLC  | = 000004 | #5-63      |         |        |        |         |        |
| ZF.LMC  | = 000100 | #5-63      |         |        |        |         |        |
| ZF.MAN  | = 020000 | #5-63      |         |        |        |         |        |
| ZF.MFL  | = 000010 | #5-63      |         |        |        |         |        |
| ZF.MTM  | = 000400 | #5-63      |         |        |        |         |        |
| ZF.MUX  | = 000040 | #5-63      |         |        |        |         |        |
| ZF.PSE  | = 002000 | #5-63      |         |        |        |         |        |
| ZF.SLI  | = 010000 | #5-63      |         |        |        |         |        |
| ZF.TIM  | = 000200 | #5-63      |         |        |        |         |        |
| ZF.X3P  | = 000000 | #5-63      |         |        |        |         |        |
| ZS.ASN  | = 100000 | #5-63      |         |        |        |         |        |
| ZS.BSY  | = 140000 | #5-63      |         |        |        |         |        |
| Z.AVL   | 000014   | #5-63      |         |        |        |         |        |
| Z.DAT   | 000016   | #5-63      | 12-299  |        |        |         |        |
| Z.DSP   | 000000   | #5-63      | 5-63    |        |        |         |        |
| Z.FLG   | 000010   | #5-63      |         |        |        |         |        |
| Z.LEN   | = 000016 | #5-63      |         |        |        |         |        |
| Z.LLN   | 000006   | #5-63      |         |        |        |         |        |
| Z.MAP   | 000020   | #5-63      |         |        |        |         |        |
| Z.NAM   | 000004   | #5-63      |         |        |        |         |        |
| Z.PCB   | 000012   | #5-63      |         |        |        |         |        |
| Z.SCH   | 000007   | #5-63      |         |        |        |         |        |
| \$MDCIN | 000244   | RG #12-296 |         |        |        |         |        |
| \$PABO  | 000216   | RG 10-248  | #11-274 | 22-643 |        |         |        |
| \$SCHPR | 001100   | RG 11-278  | 14-395  | 17-516 | 23-667 | #24-692 |        |
| \$SCHP2 | 001106   | RG #24-694 |         |        |        |         |        |

AXSCH      CREATED BY MACRO ON 28-JUN-85 AT 18:30      PAGE 2      J 4  
SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL  | VALUE      | REFERENCES        |
|---------|------------|-------------------|
| Z.LLN   | 000006     | #5-62             |
| Z.MAP   | 000020     | #5-62             |
| Z.NAM   | 000004     | #5-62             |
| Z.PCB   | 000012     | #5-62             |
| Z.SCH   | 000007     | #5-62             |
| \$FRKHD | = ***** GX | 6-203      *6-204 |
| \$SQSRV | 000000 RG  | #6-96             |



```

399 000534 016666 000010 000014 MOV 10(SP),14(SP)
400
401 000542 016666 000006 000012 50$: MOV 6(SP),12(SP) ; shuffle around stack
402 000550 016666 000004 000010 MOV 4(SP),10(SP) ; ...
403 000556 012666 000002 MOV (SP)+,2(SP) ; ...
404 000562 012666 000002 MOV (SP)+,2(SP) ; ... and clean addresses off
405
406 .IFF
407
408 MOV 4(SP),R0 ; get block address
409 MOV (R0),-(SP) ; save address of next block in list
410 MOV 10(SP), (R0) ; set new link word into block
411 MOV (R0),R0 ; point to next block
412 MOV (SP)+,(R0) ; set new link word
413 MOV (SP)+,R0 ; restore R0
414 MOV (SP)+,(SP) ; clean off stack (saving return
415 MOV (SP)+,(SP) ; ... address)
416
417 .ENDC
418
419 000566 RETURN
420

```

```

64 .SBTTL ONCE-PER-SECOND TIMER SERVICE
65 ;+
66 ;***PR1SC-ONCE-PER-SECOND TIMER SERVICE
67 ;
68 ; THIS ROUTINE IS ENTERED ONCE EVERY SECOND AND WILL DISPATCH
69 ; TIMEOUT CALLS TO COMM/EXEC PROCESSES.
70 ; -
71
72 000000 PR1SC:: .IF DF R$$MPL
73 .IF NDF R$$PRO
74
75 BIT #F2.MP,@FMSK2 ; IS THIS A MULTI-PROCESSOR?
76 BEQ 10$; BR IF NO
77 CLR R5 ; SHOW GENERAL TIMER SCAN
78
79 10$: .ENDC
80 .ENDC
81
82 000000 CALL ONESEC ; PERFORM ONCE-PER-SECOND OPERATIONS
83 000004 005167 000044 COM TIMFLG ; TOGGLE EVERY SECOND
84 000010 001010 BNE $T1SIN ; ONLY DO NEXT INSTRUCTION EVERY 2 SECONDS
85 000012 005277 000000G INC @ZTIM2 ; INCREMENT SECS/2 SINCE MIDNIGHT
86 000016 022777 124300 000000G CMP #43200.,@ZTIM2 ; WRAPAROUND?
87 000024 001002 BNE $T1SIN ; IF NOT - BR
88 000026 005077 000000G CLR @ZTIM2 ; ELSE - ZERO
89
90 000032 016700 000000G $T1SIN::MOV T1SCL,R0 ; GET ADDRESS OF CLOCK QUEUE ENTRY
91 000036 017702 000000G MOV @TKPS,R2 ; GET # OF TICKS PER SECOND
92
93 000042 005001 CLCINS: CLR R1 ; NO HIGH ORDER NUMBER OF TICKS
94
95 .IF DF R$$MPL
96 MOV #C.SYTK,R4 ; SET SYSTEM REQUEST INDICATOR
97 .IFF
98 000044 012704 000006 MOV #C.SYST,R4 ; SET SYSTEM REQUEST INDICATOR
99 .ENDC
100
101 000050 CALLR @CLINS ; AND INSERT ENTRY INTO CLOCK QUEUE
102
103 ;+
104 ; LOCAL TIMER FLAG
105 ; -
106 000054 TIMFLG: .BLKW 1

```

AXTIM CREATED BY MACRO ON 28-JUN-85 AT 18:31 PAGE 2 J 7  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL | VALUE      | REFERENCES                                           |
|--------|------------|------------------------------------------------------|
| ONESEC | 000056 R   | 6-82 #8-163                                          |
| PDDSP  | = ***** GX | 11-397                                               |
| PDSP   | = ***** GX | 11-408                                               |
| PDVNM  | = ***** GX | 8-230                                                |
| PDVTA  | = ***** GX | 8-229 8-256 11-393                                   |
| PR1SC  | 000000 RG  | #6-72                                                |
| PR100M | 000244 RG  | #9-291                                               |
| PR7    | = ***** GX | 11-347 12-430                                        |
| PS     | = ***** GX | *11-373 11-416 12-430 *12-430 *12-455                |
| PWRF1  | = ***** GX | 8-163 11-344                                         |
| R\$MPL | = *****    | 6-72 6-95 7-108 8-166 8-180 8-197 8-238 8-264 10-303 |
|        |            | 11-349 11-363 12-432 12-445 13-471 13-490            |
| R\$PRO | = *****    | 8-263                                                |
| R\$11D | = *****    | 5-60                                                 |
| R\$11M | = 000000   | 5-60                                                 |
| R\$11S | = *****    | 5-60                                                 |
| SCANST | 000300 R   | 9-292 #11-344                                        |
| SF.ACT | = 000200   | #5-59                                                |
| SF.ENA | = 000100   | #5-59                                                |
| SF.LPB | = 000004   | #5-59                                                |
| SF.MFL | = 000040   | #5-59                                                |
| SF.PAC | = 000020   | #5-59                                                |
| SF.REA | = 000010   | #5-59                                                |
| SF.SER | = 000001   | #5-59                                                |
| SF.SVC | = 000002   | #5-59                                                |
| SF.UNL | = 000040   | #5-59                                                |
| SLTMA  | = ***** GX | 8-176                                                |
| SLTNM  | = ***** GX | 8-177                                                |
| STALT  | 000476 R   | 11-384 11-411 #12-429 13-484                         |
| STDD1  | = ***** GX | 11-406                                               |
| STD11  | = ***** GX | 11-403                                               |
| STMFC  | = ***** GX | 11-396 11-407                                        |
| S.COST | 000001     | #5-59                                                |
| S.FLG  | 000000     | #5-59                                                |
| S.LEN  | 000004     | #5-59                                                |
| S.NMST | 000002     | #5-59                                                |
| S.OWNR | 000003     | #5-59                                                |
| TIMFLG | 000054 R   | *6-83 #6-106                                         |
| TKPS   | = ***** GX | 6-91                                                 |
| TK100  | = ***** GX | 9-298                                                |
| TSTIM  | = ***** GX | 8-212 8-219                                          |
| T1SCL  | = ***** GX | 6-90                                                 |
| T100C  | = ***** GX | 9-297                                                |
| T100Q  | = ***** GX | 9-291 9-294 13-469                                   |
| XF.CAN | = 000200   | #5-62                                                |
| XF.DDM | = 000001   | #5-62                                                |
| XF.DLC | = 000002   | #5-62 11-400                                         |
| XF.LLC | = 000004   | #5-62 11-389                                         |
| X\$MCB | = *****    | 5-60 5-60                                            |
| X.FLAG | 000003     | #5-62 11-382                                         |
| X.ID   | 000002     | #5-62 11-387                                         |
| X.LINK | 000000     | #5-62                                                |

J 8

SESCON - Session control connec MACRO V05.03b Friday 28-Jun-85 19:53 Page 10-2  
 Process incoming connect request

|     |        |        |        |        |              |                                                 |
|-----|--------|--------|--------|--------|--------------|-------------------------------------------------|
| 318 |        |        |        | .IF    | NDF R\$\$PRO | ; Dont retry if P/OS login failed.              |
| 319 | 000662 | 112764 | 000201 | 000010 | MOVB         | #NT.CON!CM.CON,C.FNC(R4)                        |
| 320 | 000670 | 016403 | 000026 |        | MOV          | C.BUF2+2(R4),R3 ; Recover pointer to task name  |
| 321 | 000674 | 105764 | 000033 |        | TSTB         | C.FLG2+1(R4) ; Verification level other than 0? |
| 322 | 000700 | 001361 |        |        | BNE          | 50\$ ; If NE, yes ... retry the connect         |
| 323 |        |        |        |        | .ENDC        | ; NDF R\$\$PRO                                  |
| 324 |        |        |        |        |              |                                                 |
| 325 | 000702 | 012765 | 000042 | 000022 | MOV          | #ER\$ACC,N\$ERRC(R5)                            |
| 326 |        |        |        |        |              |                                                 |
| 327 |        |        |        |        | .ENDC        |                                                 |
| 328 |        |        |        |        |              |                                                 |
| 329 | 000710 | 000261 |        | 100\$: | SEC          | ; Indicate connect error                        |
| 330 | 000712 |        |        | 110\$: | RESRG        | <R3> ; Restore register                         |
| 331 | 000714 |        |        |        | RETURN       |                                                 |

```

61 .SBTTL Network control QIO processing
62 ;+
63 ;**-$CTQIO-Network control QIO processing
64 ;
65 ; This routine process all network control QIO's.
66 ;
67 ; Inputs:
68 ; R1 = Address of the task's header
69 ; R2 = I/O subfunction code/4
70 ; R3 = Address of I/O packet
71 ; R4 = Address of mailbox
72 ; R5 = Address of database descriptor
73 ;
74 ; Outputs: (to processing routines)
75 ; R1 = Address of the task's header
76 ; R2 = Subfunction code
77 ; R3 = Address of I/O packet
78 ; R4 = Address of mailbox
79 ; R5 = Address of database descriptor
80 ;
81 .PSECT $HIGH
82
83 $CTQIO::CALLR @CTLDSP(R2) ; Dispatch to processing routine
84
85 ;+
86 ; Network control subfunction dispatch
87 ;
88
89 CTLDSP: .WORD OPN ; Network access (sub fcn= 0)
90 .WORD CLS ; Network deaccess (sub fnc= 10)
91 .WORD SPA ; Specify network data AST (sub fcn= 20)
92 .WORD GND ; Get network data (sub fcn=30)
93 .WORD GND ; Get network data length and type (sub fcn=40)
94 .WORD GND ; Get network data specific type (sub fcn=50)
95
96 .IF DF N$$PEM!N$$ACC
97 000020 000350' .WORD PEM ; Post event on mailbox
98 .IFF
99 .WORD .+1 ; Unsupported (filtered out by driver)
100 .ENDC
101
102 000022 001232' .WORD $LN ; Get local node information

```

```

553 .SBITL Network event data processing
554
555 **-EVT-Network event data processing
556
557 Process the network event.
558
559 Inputs:
560 R1 = Address of CCB
561 R3 = Address of I/O packet
562 R4 = Virtual address of user's buffer
563 User's buffer is mapped
564 R5 = Address of database descriptor
565 2(SP) - # of bytes to be transferred
566 4(SP) - High byte: Data type code
567 Low byte: I/O status return code
568 6(SP) - Address of I/O packet
569
570 Registers modified:
571 R0, R2, R4
572
573 000576 016600 000002 EVT: MOV 2(SP),R0 ; Get # of bytes to transfer
574 000602 001402 BEQ 10$; If EQ, none
575
576 000604 CALL MOVDAT ; Copy the event data
577 000610 010104 MOV R1,R4 ; Copy CCB address
578 000612 CALLR RETRES ; Return the resources

```

SESC111S CREATED BY MACRO ON 28-JUN-85 AT 19:54 PAGE 4 J 11

MACRO CROSS REFERENCE

CREF 04.00

MACRO NAME REFERENCES

|         |                           |                           |                           |                           |                           |                           |                 |                  |                  |                  |
|---------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|-----------------|------------------|------------------|------------------|
| CALL    | 8-157<br>14-510<br>18-694 | 9-192<br>15-545<br>19-722 | 9-225<br>16-576<br>19-738 | 9-229<br>17-602<br>19-740 | 9-236<br>18-643<br>19-769 | 9-240<br>18-654<br>19-773 | 9-282<br>18-678 | 11-346<br>18-685 | 11-373<br>18-688 | 13-481<br>18-691 |
| CALLE   | #6-44                     | 9-232                     |                           |                           |                           |                           |                 |                  |                  |                  |
| CALLR   | 7-83                      | 8-167                     | 9-283                     | 10-328                    | 11-377                    | 14-513                    | 15-549          | 16-578           | 18-692           | 20-826           |
| CALLX   | #9-232                    | 9-232                     |                           |                           |                           |                           |                 |                  |                  |                  |
| CCBDF\$ | #6-46                     | 6-49                      |                           |                           |                           |                           |                 |                  |                  |                  |
| CNBDF\$ | #6-46                     | 6-53                      |                           |                           |                           |                           |                 |                  |                  |                  |
| DHBDF\$ | #6-47                     | 6-57                      |                           |                           |                           |                           |                 |                  |                  |                  |
| ECDDB\$ | #6-46                     | 6-50                      |                           |                           |                           |                           |                 |                  |                  |                  |
| LLTDF\$ | #6-47                     | 6-56                      |                           |                           |                           |                           |                 |                  |                  |                  |
| LLWDF\$ | #6-46                     | 6-52                      |                           |                           |                           |                           |                 |                  |                  |                  |
| MAP     | #6-44                     | 11-370                    | 18-660                    | 19-717                    | 20-799                    |                           |                 |                  |                  |                  |
| MAPLLT  | #6-44                     |                           |                           |                           |                           |                           |                 |                  |                  |                  |
| MBXDF\$ | #6-46                     | 6-51                      |                           |                           |                           |                           |                 |                  |                  |                  |
| MSGDF\$ | #6-46                     | 6-55                      |                           |                           |                           |                           |                 |                  |                  |                  |
| NSSYM\$ | #6-46                     | 6-54                      |                           |                           |                           |                           |                 |                  |                  |                  |
| RECMAP  | #6-44                     | 18-682                    |                           |                           |                           |                           |                 |                  |                  |                  |
| RESMAP  | #6-44                     |                           |                           |                           |                           |                           |                 |                  |                  |                  |
| RESRG   | #6-44                     | 8-158                     | 9-238                     | 9-269                     | 13-482                    | 17-603                    | 18-655          |                  |                  |                  |
| RETURN  | 12-444                    | 13-485                    | 17-604                    | 19-775                    |                           |                           |                 |                  |                  |                  |
| SAVMAP  | #6-44                     |                           |                           |                           |                           |                           |                 |                  |                  |                  |
| SAVRG   | #6-44                     | 8-154                     | 9-188                     | 9-222                     | 13-477                    | 17-594                    | 18-647          |                  |                  |                  |
| SOB     | 20-810                    |                           |                           |                           |                           |                           |                 |                  |                  |                  |
| \$IERRC | #6-45                     | 11-379                    | 18-641                    | 18-645                    | 18-695                    | 19-777                    |                 |                  |                  |                  |

SESECTR11S CREATED BY MACRO ON 28-JUN-85 AT 19:54

PAGE 3 J 12

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL | VALUE | REFERENCES |
|--------|-------|------------|
|--------|-------|------------|

|           |        |       |
|-----------|--------|-------|
| E\$EVTS   | 000000 | #6-51 |
| E\$LCN    | 000016 | #6-51 |
| E\$LIN    | 000000 | #6-51 |
| E\$MOD    | 000012 | #6-51 |
| E\$NBR    | 000014 | #6-48 |
| E\$NBS    | 000020 | #6-48 |
| E\$NCR    | 000034 | #6-48 |
| E\$NCS    | 000036 | #6-48 |
| E\$NIC    | 000044 | #6-48 |
| E\$NLEN   | 000050 | #6-48 |
| E\$NLLA   | 000012 | #6-48 |
| E\$NLNK   | 000000 | #6-48 |
| E\$NML    | 000040 | #6-48 |
| E\$NMR    | 000024 | #6-48 |
| E\$NMS    | 000030 | #6-48 |
| E\$NNOD   | 000002 | #6-48 |
| E\$NOD    | 000010 | #6-51 |
| E\$NRT    | 000042 | #6-48 |
| E\$NRTP   | 000005 | #6-48 |
| E\$NSEG   | 000010 | #6-48 |
| E\$NTIM   | 000046 | #6-48 |
| E\$NUSE   | 000004 | #6-48 |
| E\$PORT   | 000014 | #6-51 |
| E\$PRM    | 000002 | #6-51 |
| E\$STAT   | 000006 | #6-51 |
| E\$STRT   | 000006 | #6-48 |
| E\$TCB    | 000004 | #6-51 |
| E\$CTL    | 000020 | #6-51 |
| E\$DATA   | 000046 | #6-51 |
| E\$EVT    | 000002 | #6-51 |
| E\$LCN    | 000042 | #6-51 |
| E\$LEN    | 000216 | #6-51 |
| E\$LIN    | 000024 | #6-51 |
| E\$LNK    | 000000 | #6-51 |
| E\$MOD    | 000036 | #6-51 |
| E\$NOD    | 000034 | #6-51 |
| E\$PDV    | 000021 | #6-51 |
| E\$PORT   | 000040 | #6-51 |
| E\$PRM    | 000026 | #6-51 |
| E\$PVC    | 000044 | #6-51 |
| E\$SIZ    | 000022 | #6-51 |
| E\$TIME   | 000004 | #6-51 |
| FR\$BCC = | 000007 | #6-51 |
| FR\$CCF = | 000001 | #6-51 |
| FR\$CDF = | 000002 | #6-51 |
| FR\$DAO = | 000011 | #6-51 |
| FR\$EXC = | 000000 | #6-51 |
| FR\$FRM = | 000010 | #6-51 |
| FR\$FTL = | 000005 | #6-51 |
| FR\$OPN = | 000004 | #6-51 |
| FR\$RFD = | 000006 | #6-51 |
| FR\$SBU = | 000012 | #6-51 |

\*9-184 \*9-187



|                   |                  |                  |                   |                   |
|-------------------|------------------|------------------|-------------------|-------------------|
| A\$LOCB= 000256RG | KISAR6= ***** GX | N.RND 000000     | X\$SDBT= 000000   | \$MAIBX 000016RG  |
| A\$CHK= 000000    | K\$AR6 000250RG  | N.RNM 000016     | ZTIM2 000366RG    | \$MENU 000030RG   |
| A\$CPS= 000000    | K\$CNT= 177546   | N.RNMC 000014    | \$ACCN 000202RG   | \$MVFBF= ***** GX |
| A\$PRI= 000000    | K\$CSR= 177546   | N.ROT 000007     | \$ALOCB= ***** GX | \$NMCLH= ***** GX |
| A\$TRP= 000000    | K\$LDC= 000000   | N.RPS 000056     | \$BLXIO= ***** GX | \$OBJHD= ***** GX |
| BLXIO 000260RG    | K\$TPS= 000074   | N.RPSC 000054    | \$BYTE 000020RG   | \$OPDAT 000226RG  |
| CALLX 000312RG    | LDBGT 000342RG   | N.RQL = 000110   | \$CALLX= ***** GX | \$OPLNG 000224RG  |
| CCBGT 000314RG    | LDBRT 000344RG   | N.RUS 000012     | \$CCBGT= ***** GX | \$PASSW 000170RG  |
| CCBRT 000316RG    | LDLPL = 000000   | OBJHD 000354RG   | \$CCBRT= ***** GX | \$PDVID= ***** GX |
| CEACC 000320RG    | LLCRS 000346RG   | PDVID 000356RG   | \$CEACC= ***** GX | \$PDVTA= ***** GX |
| CEDIV 000324RG    | LOCAL 000062RG   | PDVTA 000360RG   | \$CEDIV= ***** GX | \$QRMVF= ***** GX |
| CELOG 000322RG    | L\$ASG= 000000   | P\$P45= 000000   | \$CELOG= ***** GX | \$QUEBF= ***** GX |
| CEMUL 000326RG    | L\$DRV= 000000   | P\$WRD= 000000   | \$CEMUL= ***** GX | \$RCCB 000022RG   |
| CMPDV 000330RG    | L\$P11= 000001   | QRMVF 000274RG   | \$CMPDV= ***** GX | \$RDBRT= ***** GX |
| CSBGT 000332RG    | L\$11R= 000000   | QUEBF 000276RG   | \$CNBLK 000070RG  | \$RDBSZ= ***** GX |
| CSBRT 000334RG    | MVFBF 000350RG   | Q\$DPT= 000010   | \$CSBGT= ***** GX | \$REASN 000034RG  |
| C\$DRE= 000400    | M\$CRB= 000124   | RDBRT 000362RG   | \$CSBRT= ***** GX | \$REQID 000146RG  |
| C\$RSH= 177564    | M\$CRX= 000000   | RDBSZ 000364RG   | \$DEACB= ***** GX | \$RQCPY 000014RG  |
| DEACB 000262RG    | M\$FCS= 000000   | R\$DER= 000000   | \$DECPT= ***** GX | \$RQNAM 000010RG  |
| DECPT 000336RG    | M\$MGE= 000000   | R\$K11= 000001   | \$DEVHD= ***** GX | \$RQTCB 000006RG  |
| DEVHD 000264RG    | M\$MUP= 000000   | R\$SND= 000000   | \$DREXT= ***** GX | \$SEGMT 000026RG  |
| DREXT 000266RG    | M\$NET= 000000   | R\$11M= 000000   | \$DSDFM 000076RG  | \$SESDB 000000RG  |
| D\$BUG= 177514    | M\$OVR= 000000   | R\$11S= 000000   | \$DSASC 000100RG  | \$SESPD 000002RG  |
| D\$ISK= 000000    | NMCLH 000352RG   | SRSTD 000300RG   | \$DSNOD 000070RG  | \$SRDFM 000122RG  |
| D\$11= 000001     | N\$ACC= 000001   | STPCT 000302RG   | \$DSOBJ 000077RG  | \$SRDSC 000124RG  |
| D\$YNC= 000000    | N\$EVL= 000001   | S\$WRG= 000000   | \$ENDCD 000036RG  | \$SROBJ 000123RG  |
| D\$YNM= 000000    | N\$LDV= 000001   | S\$YSZ= 007600   | \$FVSC= ***** GX  | \$SRSTD= ***** GX |
| EVDSC 000340RG    | N\$MLL= 000001   | TKTCB 000304RG   | \$FLAGS 000047RG  | \$SRVCS 000024RG  |
| E\$XPR= 000000    | N\$MDV= 000010   | TLGTH 000252RG   | \$FLOW 000046RG   | \$STPCT= ***** GX |
| F\$ASK= 000270RG  | N\$NCT= 000001   | TSKRT 000306RG   | \$FMASK= ***** GX | \$TKTCB= ***** GX |
| F\$LVL= 000001    | N\$PEM= 000001   | TTNS 000310RG    | \$INFO 000025RG   | \$TSKRT= ***** GX |
| G\$TPP= 000000    | N.RAC 000070     | T\$KMG= 000000   | \$IOFIN= ***** GX | \$TTNS = ***** GX |
| G\$TSS= 000000    | N.RACC 000066    | T\$MIN= 000000   | \$IDPKT 000042RG  | \$UCB 000004RG    |
| G\$TTK= 000000    | N.RDE 000012     | T.LGTH= ***** GX | \$LADDR 000032RG  | \$VECLN= 000051 G |
| G\$WRD= 000000    | N.RDEC 000010    | UISAR6= ***** GX | \$LDBGT= ***** GX | \$VECTB 000246RG  |
| HOST 000054RG     | N.RFM 000006     | USAR6 000254RG   | \$LDBRT= ***** GX | \$WBLK 000040RG   |
| IOFIN 000272RG    | N.RGP 000010     | V\$YNAM 000050RG | \$LLCRS= ***** GX | \$WORK 000100RG   |
| I\$RAR= 000000    | N.RID 000034     | V\$CTR= 001000   | \$LTM 000044RG    | \$ZTIM2= ***** GX |
| I\$RDN= 000000    | N.RIDC 000032    |                  |                   |                   |

. ABS. 000110 000 (RW,I,GBL,ABS,DVR)  
000370 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

### \*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 9468 Words ( 37 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:08.37  
SY:SEDAT11S.V2,[131,134]SEDAT11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCS/PA:1,[131,10]V2.SEDAT

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE       | REFERENCES  |
|---------|-------------|-------------|
| ABT     | 000016 R    | 6-80 #7-107 |
| ACCLLT  | = ***** GX  | 7-119       |
| BRKLNK  | = ***** GX  | 8-147       |
| CPYOPT  | = ***** GX  | 7-115 8-142 |
| DISTBL  | 000014 R    | 6-71 #6-79  |
| DSC     | 000000 R    | 6-79 #7-102 |
| ER\$ABT | = 000011    | 7-107       |
| FLSHIO  | = ***** GX  | 7-110       |
| FLSHMB  | = ***** GX  | 7-113       |
| IE.BAD  | = ***** GX  | 8-151       |
| IOERR   | = ***** GX  | 8-151       |
| IOREDO  | = ***** GX  | 7-105       |
| IOSUC   | = ***** GX  | 6-73 8-149  |
| I.PRM   | = ***** GX  | 7-116 8-143 |
| M.USE   | 000010      | *8-141      |
| N\$EVL  | = 000001    | #4-2        |
| N\$SES  | = 000001    | #5-52       |
| N\$VCT  | = *****     | 7-119       |
| REJ     | 000076 R    | 6-81 #8-137 |
| SAVOPT  | = ***** GX  | 8-144       |
| SNDDIS  | = ***** GX  | 7-121       |
| TLACHK  | = ***** GX  | 8-138       |
| W.LLT   | 000004      | 7-118       |
| W.LUN   | 000003      | 7-111       |
| W.MBOX  | 000012      | 7-112       |
| W.SNDQ  | 000016      | 7-103       |
| W.TMP   | 000010      | *7-109      |
| \$CALLX | = ***** GX  | 7-119       |
| \$DSQIO | = 000000 RG | #6-69       |
| \$MAIBX | = ***** GX  | *8-137      |
| \$REASN | = ***** GX  | *7-102      |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE    | REFERENCES              |
|----------|----------|-------------------------|
| NF\$SHU  | = 000004 | 7-69 7-75 7-102         |
| NS\$DON  | = 000000 | 8-141 8-142 8-146       |
| NS\$FLG  | 000005   | *7-68 *7-69 *7-75 7-102 |
| NS\$GENQ | 000052   | 7-94 *7-96              |
| NS\$LVC  | 000036   | 8-120 8-121             |
| NS\$BUF  | = *****  | 7-77                    |
| NS\$ECL  | = *****  | 7-74                    |
| NS\$EVL  | = 000001 | #4-2 7-74               |
| NS\$SES  | = 000001 | #6-52 7-74              |
| NS\$VCT  | = *****  | 8-126                   |
| OP\$INI  | = 000000 | #6-48                   |
| OP\$TER  | = 000001 | #6-48                   |
| PH\$HDE  | = 000004 | #6-48                   |
| PH\$LOC  | = 000002 | #6-48                   |
| PH\$MTS  | = 000003 | #6-48                   |
| PH\$UMP  | = 000000 | #6-48                   |
| PH\$WCS  | = 000001 | #6-48                   |
| RE\$JECT | = *****  | 7-99                    |
| RE\$ADC  | = 000004 | #6-48                   |
| RE\$ADF  | = 000017 | #6-48                   |
| RE\$ADR  | = 000007 | #6-48                   |
| RE\$BLK  | = 000010 | #6-48                   |
| RE\$CAF  | = 000014 | #6-48                   |
| RE\$DAT  | = 000001 | #6-48                   |
| RE\$DRP  | = 000016 | #6-48                   |
| RE\$LDI  | = 000013 | #6-48                   |
| RE\$LSN  | = 000012 | #6-48                   |
| RE\$NML  | = 000001 | #6-48                   |
| RE\$OPE  | = 000004 | #6-48                   |
| RE\$OPR  | = 000000 | #6-48 7-74              |
| RE\$RCV  | = 000001 | #6-48                   |
| RE\$SED  | = 000011 | #6-48                   |
| RE\$SKW  | = 000006 | #6-48                   |
| RE\$STA  | = 000002 | #6-48                   |
| RE\$SUM  | = 000003 | #6-48                   |
| RE\$SYN  | = 000000 | #6-48                   |
| RE\$TME  | = 000021 | #6-48                   |
| RE\$TMO  | = 000000 | #6-48                   |
| RE\$TMR  | = 000020 | #6-48                   |
| RE\$UPT  | = 000002 | #6-48                   |
| RE\$URE  | = 000003 | #6-48                   |
| RE\$VER  | = 000005 | #6-48                   |
| RE\$VRO  | = 000015 | #6-48                   |
| RT\$INI  | = 000002 | #6-48                   |
| RT\$OFF  | = 000001 | #6-48                   |
| RT\$ON   | = 000000 | #6-48                   |
| SC\$OFF  | = 000001 | #6-48                   |
| SC\$ON   | = 000000 | #6-48 7-74              |
| SC\$RST  | = 000003 | #6-48                   |
| SC\$SHU  | = 000002 | #6-48 7-74              |
| SV\$DUM  | = 000001 | #6-48                   |
| SV\$LDA  | = 000000 | #6-48                   |

SESDSP - Session control dispat MACRO V05.03b Friday 28-Jun-85 19:56 J.16 Page 10-1  
Request task to run (from AUX request)

|     |        |        |       |         |                                          |
|-----|--------|--------|-------|---------|------------------------------------------|
| 328 | 000340 | 001313 | BNE   | 10\$    | ; If NE, yes - check next chain          |
| 329 | 000342 | 011404 | MOV   | (R4),R4 | ; Else, skip over the rest of this chain |
| 330 | 000344 | 001401 | BEQ   | 50\$    | ; Just in case                           |
| 331 | 000346 | 000771 | BR    | 40\$    | ; Continue with this chain               |
| 332 | 000350 |        | 50\$: |         |                                          |
| 333 | 000350 |        | RET:  | RETURN  | ; Main line SWSTK return point           |
| 334 |        |        |       |         |                                          |

```

106 .SBTTL TIMER SERVICE - STATE DISPATCH TABLE
107
108 ;*-MDMSRV-STATE DISPATCH TABLE
109 :
110 : THIS TABLE IS USED TO DISPATCH PROCESSING ON EACH ACTIVE LINE
111 : ON THE ONCE-PER-SECOND TIMER.
112 :
113 :
114 MDMSRV: .WORD .+1 ; IDLE LINE (SHOULD NEVER DISPATCH HERE)
115 .WORD .WRNG ; WAITING FOR RING
116 .WORD .WCON ; WAITING FOR CONNECT
117 .WORD .COND ; CARRIER DELAY TO SEE IF CONNECTED
118 .IF DF X$$$D52
119 .WORD .COND2 ; Waiting for Carrier and Clear To Send
120 .WORD .CONFL ; Connect failure
121 .ENDC
122 .WORD .ACTV ; LINE IS ACTIVE
123 .WORD .ACTD ; CARRIER DELAY TO SEE IF LINE IS ACTIVE
124 .WORD .CONE ; LOST CARRIER OR DSR WHILE ACTIVE
125 .WORD .DISP ; DISCONNECT DELAY TO ALLOW CIRCUITS TO RESET
126 .WORD .PABU ; POST ABORT TO DLC LEVEL
127

```

```

502 .SBTTL CNCMP - COMPLETE THE CONNECTION
503
504 :+
505 :*-CNCMP-COMPLETE THE CONNECTION
506 THE MODEM IS NOW READY TO USE. POST A COMPLETION THE THE DLC LEVEL.
507 :-
508 INPUTS:
509 R2 - POINTER TO SYSTEM LINE TABLE
510 R4 - SYSTEM LINE NUMBER
511 R5 - POINTER TO LINE ENTRY IN MDC DATABASE
512
513 000610 012703 000020 CNCMP: MOV #CS.ENB,R3 ; SET SUCCESSFUL STATUS
514 000614 112702 000010 : MOVB #ST.ACT,R2 ; SET NEW STATE ... LINE ACTIVE
515 000620 012701 012000 : MOV #FS.ENB,R1 ; SET SUBFUNCTION CODE
516 000624 : CALL $SCHPR ; AND SCHEDULE DLC PROCESS
517 000630 103003 : BCC 10$; CCB ALLOCATION SUCCESSFUL?
518 000632 152765 000200 000004 : BISB #MC.CCB, M.CSV(R5) ; NO ... MARK ALLOCATION FAILURE
519 000640 : 10$: RETURN
520 ;

```

AXMDC      CREATED BY MACRO ON 28-JUN-85 AT 18:30      PAGE 4      K 3  
SYMBOL CROSS REFERENCE      CREF 04.00

| SYMBOL | VALUE    | REFERENCES           |
|--------|----------|----------------------|
| .ACTD  | 000720 R | 6-123 #19-580        |
| .ACTV  | 000642 R | 6-122 #18-535        |
| .COND  | 000554 R | 6-117 #16-451        |
| .CONE  | 001006 R | 6-124 #21-625        |
| .DISD  | 001040 R | 6-125 22-644 #23-662 |
| .ABO   | 001022 R | 6-126 #22-643        |
| .WCON  | 000500 R | 6-116 #15-410        |
| .WRNG  | 000446 R | 6-115 #14-390        |

AXSCH      CREATED BY    MACRO    ON 28-JUN-85 AT 18:30      PAGE 3      K 4  
MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES                             |
|------------|----------------------------------------|
| CALL       | 6-178      6-185      6-186            |
| CCBDF\$    | #5-59      5-60                        |
| ENABL\$    | #5-58                                  |
| INHIB\$    | #5-58                                  |
| MIPS       | 6-100      6-156      6-187      6-227 |
| OPTDF\$    | #5-59      5-61                        |
| PDVDF\$    | #5-59      5-62                        |
| RESRG      | #5-58                                  |
| RETURN     | 6-228                                  |
| SAVRG      | #5-58                                  |



\$CMINI - Component buffer limit (init)

```

422 .SBTTL $CMINI - Component buffer limit (init)
423
424 ;+
425 ;***$CMINI-Component buffer limit (init)
426 ;***$CMEXI-Component buffer limit (exit)
427
428 ; This routine is called by a component (ie.DECnet,X.25) when
429 ; it is initializing or exiting. In a combined system each component is
430 ; allowed to use buffers up to the 'square root limit'.
431 ; That is (#total buffers)/(SQRT (#components)).
432 ; Each component is responsible for staying below that limit.
433 ; $CMINI is called at component startup,
434 ; $CMEXI is called at component shutdown.
435
436 ; Inputs:
437 ; None
438
439 ; Outputs:
440 ; $SQRCM (byte) is updated with the new buffer
441 ; limit for each component
442
443 ; Registers modified:
444 ; R0,R1
445
446 .ENABL LSB
447 $CMINI::MOV SQRCM,R0 ; get address of square root factor cell
448 INCB 1(R0) ; component startup entry point
449 BR 5$;
450
451 $CMEXI::MOV SQRCM,R0 ; get address of square root factor cell
452 DECB 1(R0) ; component shutdown entry point
453
454 5$: MOVB 1(R0),R1 ; copy number coresident components
455 BEQ 20$; if eq, no components left
456 MOVB $SQRTB-1(R1),R1 ; get the SQRT (r1*2) from table
457 MOV @RDBNM,R0 ; copy number of LDBs in system
458 SUB @RDBTH,R0 ; reduce it by (# reserved for device receives)
459 ASL R0 ; scale R0 (scale factor=2)
460 CALL @CEDIV ; compute new buffer limit
461 MOVB R0,@SQRCM ; and store it
462 20$: RETURN
463
464 .DSABL LSB

```

K 6

```

108 .IF DF R$$MPL
109 .IF NDF R$$PRO
110
111 ;+
112 ;**--MP1SC-ONCE-PER-SECOND TIMER SERVICE (MULTI-PROCESSOR SUPPORT)
113 ;**--$SMP1S-START PROCESSOR DEPENDANT ONE SECOND TIMER
114 ;**--$SMPTM-START PROCESSOR DEPENDANT TIMER
115
116 ; THIS ROUTINE IS ENTERED ONCE PER SECOND ON EVERY PROCESSOR WHICH
117 ; HAS ACTIVE LINES.
118 ; -
119 .ENABL LSB
120
121 MP1SC:: MOVB @PROC2,R5 ; GET PROCESSOR NUMBER *2
122 ADD .1SCTB,R5 ; GET ADDRESS OF TIMER QUEUE ELEMENT
123 MOV (R5),R5 ; ...
124 MOV #-1,-2(R5) ; SHOW TIMER CELL IS FREE
125
126 CALL ONESEC ; PERFORM ONCE-PER-SECOND OPERATIONS
127
128 TST $ACTLN ; ANY ACTIVE LINES LEFT ON THIS PROCESSOR?
129 BEQ 10$; IF EQ, NO
130
131 $SMP1S:: MOVB @PROC2,R0 ; GET PROCESSOR NUMBER *2
132 ADD .1SCTB,R0 ; GET ADDRESS OF TIMER QUEUE ELEMENT
133 MOV (R0),R0 ; ...
134 INC -2(R0) ; IS THE TIMER ALREADY RUNNING?
135 BNE 10$; IF NE, YES
136
137 MOV @TKPS,R2 ; GET # OF TICKS PER SECOND
138
139 $SMPTM:: MOV #MP1SC,C.SUB(R0); SET UP ADDRESS OF PROCESSING ROUTINE
140 MOV @CPBIT,C.URM(R0); SET UP URM OF DESTINATION PROCESSOR
141
142 CLR R1 ; NO HIGH ORDER TICKS
143 MOV #C.SYST!100000,R4 ; SET SYSTEM REQUEST INDICATOR
144 CALLR @CLINS ; INSERT ENTRY IN CLOCK QUEUE
145
146 10$: RETURN
147
148 .DSABL LSB
149 .ENDC
150 .ENDC

```

L 6

AXTIM      CREATED BY MACRO ON 28-JUN-85 AT 18:31      PAGE 3      K 7

SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL  | VALUE      | REFERENCES                                   |
|---------|------------|----------------------------------------------|
| X.RTMR  | 000006     | #5-62      11-380      13-468                |
| X.TMR   | 000004     | #5-62      *11-378      *11-380      *13-468 |
| ZF.COU  | = 001000   | #5-60                                        |
| ZF.DDM  | = 000001   | #5-60                                        |
| ZF.DIA  | = 004000   | #5-60                                        |
| ZF.DLC  | = 000002   | #5-60                                        |
| ZF.DVP  | = 100000   | #5-60                                        |
| ZF.INI  | = 040000   | #5-60                                        |
| ZF.KMX  | = 000020   | #5-60                                        |
| ZF.LLC  | = 000004   | #5-60                                        |
| ZF.LMC  | = 000100   | #5-60                                        |
| ZF.MAN  | = 020000   | #5-60                                        |
| ZF.MFL  | = 000010   | #5-60                                        |
| ZF.MTM  | = 000400   | #5-60                                        |
| ZF.MUX  | = 000040   | #5-60                                        |
| ZF.PSE  | = 002000   | #5-60                                        |
| ZF.SLI  | = 010000   | #5-60                                        |
| ZF.TIM  | = 000200   | #5-60      8-235                             |
| ZF.X3P  | = 000000   | #5-60                                        |
| ZS.ASN  | = 100000   | #5-60                                        |
| ZS.BSY  | = 140000   | #5-60                                        |
| ZTIM2   | = ***** GX | 6-85      6-86      6-88                     |
| Z.AVL   | 000014     | #5-60                                        |
| Z.DAT   | 000016     | #5-60      8-254      11-395                 |
| Z.DSP   | 000000     | #5-60                                        |
| Z.FLG   | 000010     | #5-60      8-235                             |
| Z.LEN   | = 000016   | #5-60                                        |
| Z.LLN   | 000006     | #5-60                                        |
| Z.MAP   | 000020     | #5-60                                        |
| Z.NAM   | 000004     | #5-60                                        |
| Z.PCB   | 000012     | #5-60      8-233                             |
| Z.SCH   | 000007     | #5-60                                        |
| \$STSTM | 000530 RG  | #13-467                                      |
| \$T1SIN | 000032 RC  | 6-84      6-87      #6-90                    |
| \$T100I | 000254 R   | #9-294      13-505                           |

```

333 .SBTTL Queue connect request to task
334
335 +
336 **--QUETSK-Queue connect request to task
337
338 If the destination task is already active place the connect request
339 on it's mailbox and fire off an AST. If it is not active, request
340 the task to run and queue the request on the general delivery queue.
341
342 -
343 Inputs:
344 R3 = Pointer to task name (in RAD50)
345 R4 = Address of CCB
346 R5 = Address of database descriptor
347
348 Outputs:
349 'C' Clear - Request successfully queued
350 'C' Set - Failed to queue request
351
352 Registers modified:
353 R0, R1
354
355 000716 QUETSK::.IF DF R$$MPL
356
357 CALL @SRPRO ; Scan STD for task's TCB
358
359 .IFF
360
361 000716 CALL @SRSTD ; Scan STD for task's TCB
362
363 .ENDC
364
365 .IF DF N$$SLI
366
367 BCC 10$; If CC, task found
368 MOV #ERSUOB,N$ERRC(R5) ; Reason is 'object not installed'
369
370 .IF DF N$$ACC
371
372 CMPB #NT,VFY!CM.CON,C.FNC(R4)
373 BEQ 100$; Can't verify using the SLI
374
375 .ENDC
376
377 10$: CALLR QUESLI ; Try to queue request via SLI
378
379 .IFF
380
381 000722 103443 BCS 100$; If CS, task not installed
382
383 .ENDC
384
385 000724 010501 MOV R5,R1 ; Compute address of mailbox listhead
386 000726 062701 000046 ADD #N$MBXQ-M,NEXT,R1
387 000732 016101 000002 MOV M,NEXT(R1),R1 ; Get address of next mailbox
388 000736 001432 BEQ 40$; If EQ, end of list
389 000740 020061 000004 CMP R0,M.TASK(R1) ; Is this the right mailbox?
390 000744 001372 BNE 20$; If NE, no

```

```

104 .SBTTL Network access
105
106 *
107 **--OPN-Network access
108
109 This routine establishes a mailbox for use by the network software.
110
111 Inputs:
112 R1 = Address of the task's header
113 R2 = Subfunction code/4
114 R3 = Address of I/O packet
115 R4 = Address of mailbox
116 R5 = Address of database descriptor
117
118 000024 016564 000050 000002 OPN: MOV N$MBXQ(R5),M.NEXT(R4)
119 000032 010465 000050 MOV R4,N$MBXQ(R5) ; Link mailbox into mailbox queue
120
121 000036 010502 MOV R5,R2 ; Point at general delivery queue listhead
122 000040 062702 000052 ADD #N$GENQ,R2 ; ...
123
124 000044 010400 MOV R4,R0 ; Point to network data listhead
125 000046 062700 000014 ADD #M.MAIL,R0 ; ...
126
127 000052 010203 10$: MOV R2,R3 ; Save current position in list
128 000054 011202 MOV (R2),R2 ; Get next entry in the list
129 000056 001443 BEQ 50$; If EQ, none
130
131 000060 026264 000004 000004 CMP C.NSP(R2),M.TASK(R4)
132 000066 001371 BNE 10$; If NE, not for this task
133 000070 011213 MOV (R2),(R3) ; Unlink entry from queue
134 000072 005012 CLR (R2) ; and clear link pointer
135
136 .IF DF N$SACC
137 000074 122762 000207 000010 CMPB #NT.VFY!CM.CON,C.FNC(R2)
138 000102 001423 BEQ 20$; Always allow verification requests to pass
139
140 .ENDC
141
142 .IF DF N$PEM
143
144 000104 122762 000006 000010 CMPB #NT.EVT,C.FNC(R2)
145 000112 001421 BEQ 30$; Always allow PEM$ requests to pass
146
147 .ENDC
148
149 000114 105764 000011 TSTB M.MAX(R4) ; Has user restricted max # of links?
150 000120 001414 BEQ 20$; If EQ, no
151 000122 126464 000010 000011 CMPB M.USE(R4),M.MAX(R4)
152 000130 103410 BLO 20$; If LO, we can accept this link
153
154 000132 SAVRG <R4> ; Save address of mailbox
155 000134 010204 MOV R2,R4 ; Copy CCB address
156 000136 012701 000006 MOV #ER$MLB,R1 ; Reason = object too busy
157 000142 CALL REJECT ; Reject the connection
158 000146 RESRG <R4> ; Recover mailbox address
159 000150 000404 BR 40$
160

```

K 10

```

580 .SBTTL Move data to user buffer
581 :+
582 **:MOVDAT-Move data to user buffer
583 :
584 Copy data from a mapped buffer to the user specified buffer.
585 :-
586 Inputs:
587 R0 = # of bytes to transfer
588 R1 = Address of CCB
589 R3 = Address of I/O packet
590 :
591 Registers modified:
592 R0, R2, R4
593 :
594 000616 MOVDAT: SAVRG <R1,R3> ; Save some registers
595 000622 016102 MOV C.BUF+2(R1),R2 ; Get source virtual address
596 000626 020227 CMP R2,#140000 ; Is virtual address in APR6?
597 000632 103402 BLO 10$; If LO, no
598 000634 162702 SUB #20000,R2 ; Convert to APR5
599
600 000640 016101 MOV C.BUF(R1),R1 ; Get source bias
601 000644 016303 MOV I.PRM(R3),R3 ; Get destination bias
602 000650 CALL @BLXIO ; Copy the data
603 000654 RESRG <R3,R1> ; Recover the registers
604 000660 RETURN

```

L 10

\*\*FILE\*\*ID\*\*SESCIR

```

SSSSSSSS EEEEEEEEE SSSSSSSS CCCCCCCC TTTTTTTTT RRRRRRRR
SSSSSSSS EEEEEEEEE SSSSSSSS CCCCCCCC TTTTTTTTT RRRRRRRR
SS EE SS CC TT RR RR
SS EE SS CC TT RR RR
SS EE SS CC TT RR RR
SS EE SS CC TT RR RR
SSSSSS EEEEEEEEE SSSSSS CC TT RRRRRRRR
SSSSSS EEEEEEEEE SSSSSS CC TT RRRRRRRR
 SS EE SS CC TT RR RR
 SS EE SS CC TT RR RR
 SS EE SS CC TT RR RR
SSSSSSS EEEEEEEEE SSSSSSSS CCCCCCCC TT RR RR
SSSSSSS EEEEEEEEE SSSSSSSS CCCCCCCC TT RR RR

```

```

....
....
....
....

```

```

11 11 SSSSSSSS
11 11 SSSSSSSS
1111 1111 SS
1111 1111 SS
11 11 SS
11 11 SS
11 11 SSSSSS
11 11 SSSSSS
11 11 S-
11 11 SS
11 11 SS
11 11 SS
111111 111111 SSSSSSSS
111111 111111 SSSSSSSS

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE    | REFERENCES        |
|---------|----------|-------------------|
| FR\$SHO | = 000003 | #6-51             |
| FR\$UBU | = 000013 | #6-51             |
| FR\$UPT | = 000014 | #6-51             |
| KISAR6  | = *****  | *7-74 *7-81 *7-87 |
| L.CTR   | = 000074 | 7-77              |
| MO\$SAC | = 000016 | #6-51             |
| MO\$SPR | = 000012 | #6-51             |
| MO\$SSV | = 000014 | #6-51             |
| MO\$25A | = 000006 | #6-51             |
| MO\$25P | = 000002 | #6-51             |
| MO\$25S | = 000004 | #6-51             |
| MO\$29S | = 000010 | #6-51             |
| M\$HIGH | = 000003 | #6-51             |
| M\$3100 | = 000000 | #6-51             |
| M\$3101 | = 000001 | #6-51             |
| M\$3102 | = 000002 | #6-51             |
| M\$3103 | = 000003 | 6-51              |
| N\$CIR  | = 000034 | *8-148 *8-150     |
| N\$ENC  | = 000042 | 7-81              |
| N\$LLTM | = 000024 | 7-74              |
| N\$EVL  | = 000001 | 9-157             |
| N\$MLL  | = 000001 | 7-74              |
| N\$NCT  | = 000001 | 7-55              |
| N\$SES  | = 000001 | #6-53             |
| N\$VCT  | = *****  | 7-74 7-81 7-87    |
| OP\$INI | = 000000 | #6-51             |
| OP\$TER | = 000001 | #6-51             |
| PH\$HDE | = 000004 | #6-51             |
| PH\$LOC | = 000002 | #6-51             |
| PH\$MTS | = 000003 | #6-51             |
| PH\$UMP | = 000000 | #6-51             |
| PH\$WCS | = 000001 | #6-51             |
| RE\$ADC | = 000004 | #6-51             |
| RE\$ADF | = 000017 | #6-51             |
| RE\$ADR | = 000007 | #6-51             |
| RE\$BLK | = 000010 | #6-51             |
| RE\$CAF | = 000014 | #6-51             |
| RE\$DAT | = 000001 | #6-51             |
| RE\$DRP | = 000016 | #6-51             |
| RE\$LDT | = 000013 | #6-51             |
| RE\$LSN | = 000012 | #6-51             |
| RE\$NML | = 000001 | #6-51             |
| RE\$OPE | = 000004 | #6-51             |
| RE\$OPR | = 000000 | #6-51             |
| RE\$RCV | = 000001 | #6-51             |
| RE\$SED | = 000011 | #6-51             |
| RE\$SKW | = 000006 | #6-51             |
| RE\$STA | = 000002 | #6-51             |
| RE\$SUM | = 000003 | #6-51             |
| RE\$SYN | = 000000 | #6-51             |
| RE\$TME | = 000021 | #6-51             |
| RE\$TMO | = 000000 | #6-51             |



SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL VALUE REFERENCES

|         |            |        |
|---------|------------|--------|
| ALOCB   | 000256 RG  | #9-135 |
| BLXIO   | 000260 RG  | #9-136 |
| CALLX   | 000312 RG  | #9-159 |
| CCBGT   | 000314 RG  | #9-160 |
| CCBRT   | 000316 RG  | #9-161 |
| CEACC   | 000320 RG  | #9-162 |
| CEDIV   | 000324 RG  | #9-166 |
| CELOG   | 000322 RG  | #9-163 |
| CEMUL   | 000326 RG  | #9-167 |
| CMPDV   | 000330 RG  | #9-170 |
| CSBGT   | 000332 RG  | #9-171 |
| CSBRT   | 000334 RG  | #9-172 |
| DEACB   | 000262 RG  | #9-137 |
| DECPT   | 000336 RG  | #9-173 |
| DEVHD   | 000264 RG  | #9-138 |
| DREXT   | 000266 RG  | #9-139 |
| EVDSC   | 000340 RG  | #9-174 |
| FMASK   | 000270 RG  | #9-140 |
| H0ST    | 000054 RG  | #7-105 |
| IOFIN   | 000272 RG  | #9-141 |
| KISAR6  | = ***** GX | 9-132  |
| KSAR6   | 000250 RG  | #9-132 |
| LDBGI   | 000342 RG  | #9-175 |
| LDBRT   | 000344 RG  | #9-176 |
| LLCRS   | 000346 RG  | #9-177 |
| LOCAL   | 000062 RG  | #7-106 |
| MVFBF   | 000350 RG  | #9-178 |
| NMCLH   | 000352 RG  | #9-179 |
| N\$SACC | = 000001   | 7-99   |
| N\$SEVL | = 000001   | #4-2   |
| N\$SMCP | = *****    | 9-152  |
| N\$SLI  | = *****    | 7-81   |
| OBJHD   | 000354 RG  | #9-180 |
| PDVID   | 000356 RG  | #9-181 |
| PDVTA   | 000360 RG  | #9-182 |
| QRMVF   | 000274 RG  | #9-142 |
| QUEBF   | 000276 RG  | #9-143 |
| RDBRT   | 000362 RG  | #9-183 |
| RDBSZ   | 000364 RG  | #9-184 |
| R\$SEIS | = *****    | 9-165  |
| R\$SMPL | = *****    | 9-145  |
| R\$SPRO | = *****    | 9-185  |
| SRSTD   | 000300 RG  | #9-149 |
| STPCT   | 000302 RG  | #9-150 |
| TKTCB   | 000304 RG  | #9-156 |
| TLGTH   | 000252 RG  | #9-133 |
| TSKRT   | 000306 RG  | #9-157 |
| TTNS    | 000310 RG  | #9-158 |
| T.LGTH  | = ***** GX | 9-133  |
| UISAR6  | = ***** GX | 9-134  |
| USAR6   | 000254 RG  | #9-134 |
| VFYNAM  | 000050 RG  | #7-101 |

SESDIS11S CREATED BY MACRO ON 28-JUN-85 AT 19:55 PAGE 2 K 14

MACRO CROSS REFERENCE

CREF 04.00

| MACRO NAME | REFERENCES  |
|------------|-------------|
| CALL       | 7-110       |
| CALLE      | #5-43 7-113 |
| CALLR      | 6-71 7-119  |
| CALLX      | #7-119 6-73 |
| ECDDBS     | #5-45 7-119 |
| LLTDF\$    | #5-45 5-49  |
| LLWDF\$    | #5-45 5-50  |
| MAP        | #5-43 5-47  |
| MAPLLT     | #5-43       |
| MBXDF\$    | #5-45 5-48  |
| RESRG      | #5-43       |
| SAVRG      | #5-43       |
| \$IERRC    | #5-44 8-151 |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE       | REFERENCES  |
|---------|-------------|-------------|
| US\$CNF | = 000002    | 8-141       |
| US\$DIS | = 000006    | 8-142 8-144 |
| US\$DSC | = 000004    | 8-146       |
| US.MDM  | = ***** GX  | 7-73        |
| U.STS   | = ***** GX  | *7-73       |
| WORD1   | = 000200    | #7-74 7-74  |
| WORD2   | = 000000    | #7-74 7-74  |
| \$CALLX | = ***** GX  | 8-126       |
| \$DMQIO | = 000000 RG | #7-68       |
| \$UCB   | = ***** GX  | 7-72        |

```

336
337 000316 .PSECT
338
339 .SBTTL Request task to run
340
341 +
342 ***-REQTSK-Request task to run
343
344 Check that the specified task is installed and request it to run.
345 If necessary, spawn a new copy of a multi-copy task (one whose task
346 name ends with '$$$' or whose prototype TCB is in secondary pool).
347
348 Inputs:
349 R4 = Address of CCB
350 R5 = Address of database descriptor
351
352 Outputs:
353 'C' Clear - Task requested successfully
354 'C' Set - Failed to find TCB or request task
355
356 Registers modified:
357 R0, R1, R3
358
359 000316 .PSECT
360
361 REQTSK: MOV B C.FNC(R4),R1 ; Get the function code
362 BIC #^C<17>,R1 ; Isolate useful bits
363 ASL R1 ; Form word offset
364 CALL @NDISP-2(R1) ; Get pointer to task name string
365
366 REQTS1: .IF DF R$$$MPL ; alternate entry for task requesting (AUX)
367 CALL @SRPRO ; Search prototype TCB list
368 .IFF
369 CALL @SRSTD ; Search prototype TCB list
370
371 000334 .ENDC
372
373
374 BCS 60$; If CS, task not installed
375 BICB #CX.REM,C.MOD(R4) ; Indicate no new TCB generated
376
377 MOV R0,$RQTCB ; Save address of TCB
378 000350 010067 000000G
379
380 .IF DF R$$$MPL&N$$$MCP
381
382 BIT #1,R0 ; Is TCB in secondary pool?
383 BEQ 10$; If EQ, no
384 MAP -1(R0) ; Map to the TCB
385 MOV #140000,R0 ; Set up virtual address of TCB
386 CMP T.NAM+2(R0),#^R$$$; Is it a valid multi copy installed name?
387 BNE 5$; If NE, No - allow only 1 copy
388 BITB #CX.SMC,C.MOD(R4) ; Spawn multiple copies?
389 BNE 20$; If NE, yes - leave copy count as is
390 5$: MOV #1,$RQCPY ; Else, allow only one copy
391 BR 20$; Always copy the TCB from secondary pool
392

```

129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164

000022 105715  
000024 100407  
000026 126564 000001 000006  
000034 001404  
000036 062705 000010  
000042 000767  
000044  
000046 116465 000007 000006  
000054  
  
000060 000066'  
000062 000114'  
000064 000134'

```
.SBTTL CONTROL ENABLE SERVICE
+
**--MDMCTL-CONTROL ENABLE SERVICE

THIS ROUTINE IS ENTERED FROM THE COMM/EXEC DURING CERTAIN CONTROL
ENABLE REQUESTS FROM THE DLC TO THE DDM. ON RETURN FROM THIS ROUTINE
THE CONTROL ENABLE REQUEST WILL BE ROUTED TO THE DDM PROCESS.

INPUTS:
R3 - SUB-FUNCTION CODE:
 FS.RNG - LOOK FOR RING
 FS.ENB - ENABLE LINE
 FS.DIS - DISABLE LINE
R4 - CCB
R5 - POINTER TO MDC DATABASE

OUTPUTS:
ONLY R4 WILL BE GUARANTEED TO BE PRESERVED

MDMCTL::TSTB (R5) ; HAVE WE SCANNED ALL OF THE TABLE?
BMB 10$; YES ... IGNORE THIS REQUEST
CMPB M.LIN(R5),C.LIN(R4) ; IS IT FOR THIS LINE?
BEQ 20$; YES ... DISPATCH TO PROCESSOR
ADD #M.LEN,R5 ; NO ... MOVE ON TO NEXT ENTRY
BR MDMCTL
10$: RETURN ; LINE NOT IN DATABASE ... REQUEST IGNORED
20$: MOVB C.STA(R4),M.STA(R5) ; SAVE STATION ADDRESS
CALLR @CTLDSP-<FS.RNG/400>(R3)

+
FUNCTION DISPATCH TABLE

CTLDSP: .WORD RING ; LOOK FOR RING
 .WORD ENABLE ; ENABLE LINE
 .WORD DSABLE ; DISABLE LINE
;
```

```

522 .SBTTL ST.ACT - LINE ACTIVE
523
524 **-.ACTV-LINE ACTIVE
525
526 THE LINE IS ACTIVE (IN USE BY THE NETWORK), MONITOR THE STATE OF
527 CARRIER (ASYNC) OR DATASET READY (SYNC) TO CHECK FOR POSSIBLE
528 LINE FAILURES.
529
530 INPUTS:
531 R2 - POINTER TO SYSTEM LINE TABLE
532 R4 - SYSTEM LINE NUMBER
533 R5 - POINTER TO LINE ENTRY IN MDC DATABASE
534
535 000642 032712 001000 .ACTV: BIT #LF.LPB,(R2) ; IS THIS LINE IN LODPBACK?
536 000646 001015 BNE 10$; IF NE, YES
537 .IF NDF X$$D52
538 000650 132715 000001 BITB #MS.SYN,(R5) ; IS THIS A SYNCHRONOUS LINE?
539 000654 001013 BNE 20$; YES
540 000656 132765 000001 000004 BITB #MC.CAR,M.CSV(R5) ; NO ... IS CARRIER STILL ASSERTED?
541 000664 001006 BNE 10$; YES ... LINE IS STILL ACTIVE
542 000666 112765 000002 000002 MOVB #TM.ACT,M.TIM(R5) ; NO ... START TIMER TO CHECK FOR CARRIER
543 000674 112765 000012 000003 MOVB #ST.ADL,M.STT(R5) ; SET STATE TO ACTIVE DELAY
544 000702 10$: RETURN
545 000704 132765 000002 000004 BITB #MC.DSR,M.CSV(R5) ; SYNC LINE ... IS DATASET READY ASSERTED?
546 000712 001373 BNE 10$; YES ... LINE IS STILL ACTIVE
547 000714 CALLR DISC ; POST ASYNCHRONOUS DISCONNECT
548
549 .IFF
550
551 Monitor the state of Data Set Ready.
552 If the modem generates continuous carrier, monitor Carrier Detect also.
553
554 BITB #MC.DSR, M.CSV(R5) ; If Data Set Ready not asserted
555 BNE 10$; then
556 CALLR DISC ; Goto post async disconnect
557
558 10$: BITB #MS.SWT, M.STS(R5) ; If carrier is continuous
559 BNE 20$; then
560 BITB #MC.CAR, M.CSV(R5) ; If Carrier not asserted
561 BNE 20$; then
562 MOVB #TM.ACT, M.TIM(R5) ; Start delay timer
563 MOVB #ST.ADL, M.STT(R5) ; Set state to delay
564
565 20$: RETURN
566 .ENDC

```

AXMDC      CREATED BY    MACRO    ON 28-JUN-85 AT 18:30      PAGE 5      L 3

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES |        |        |        |        |        |        |        |        |        |
|------------|------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| CALL       | 10-248     | 11-278 | 13-361 | 13-371 | 17-516 | 20-604 | 22-643 | 23-667 | 24-692 | 24-698 |
| CALLR      | #5-58      | 7-156  | 14-395 | 18-547 | 19-584 | 21-627 |        |        |        |        |
| CCBDF\$    | #5-58      | 5-64   |        |        |        |        |        |        |        |        |
| MDCDF\$    | #5-58      | 5-65   |        |        |        |        |        |        |        |        |
| PDVDF\$    | #5-58      | 5-63   |        |        |        |        |        |        |        |        |
| RETURN     | 7-154      | 8-181  | 9-219  | 10-246 | 10-256 | 11-280 | 12-308 | 13-376 | 14-396 | 15-419 |
|            | 15-422     | 16-455 | 16-458 | 17-519 | 18-544 | 19-587 | 20-607 | 22-646 | 23-670 | 24-700 |
| SLTDF\$    | #5-58      | 5-66   |        |        |        |        |        |        |        |        |

\*\*FILE\*\*ID\*\*AXSUB

L 4

```
AAAAAA XX XX SSSSSSS UU UU BBBB BBBB
AAAAAA XX XX SSSSSSS UU UU BBBB BBBB
AA AA XX XX SS UU UU BB BB
AA AA XX XX SS UU UU BB BB
AA AA XX XX SS UU UU BB BB
AA AA XX XX SS UU UU BB BB
AA AA XX XX SS UU UU BB BB
AAAAAA XX XX SS UU UU BB BB
AAAAAA XX XX SS UU UU BB BB
AA AA XX XX SS UU UU BB BB
AA AA XX XX SSSSSSS UUUUUUUUU BBBB BBBB
AA AA XX XX SSSSSSS UUUUUUUUU BBBB BBBB
```

```
LL SSSSSSS TTTTTTTTT
LL SSSSSSS TTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSS TT
LL SSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLL SSSSSSS TT
LLLLLLLL SSSSSSS TT
```



```

466 .SBTTL $RQSTL - Request System Microcode Loader
467
468 ;**-$RQSTL- Request System Microcode Loader
469
470 Function: To allocate a CCB, initialize it for a microcode load
471 request, queue it to the microcode loader, and request
472 the loader for execution.
473
474 Call Format: CALL $RQSTL
475
476 Inputs: R2 = mapping bias of an ASCII microcode filename (RSX-11M/M+)
477 = mapping bias of an ASCII microcode partition (RSX-11S)
478 R3 = address of ASCII string
479 R4 = system line number
480
481 Outputs: R3 = completion status
482 C-bit set - Request failed, R3 = CS.ERR+CS.ABO
483 C-bit clear - Request succeeded, R3 = 0
484
485 Registers: R2 and R4 destroyed
486
487 ;
488 $RQSTL::MOV R0,-(SP) ; Save a register
489 MOV R1,-(SP) ; Save a register
490 MOV R3,-(SP) ; Save address of input string
491
492 492 000650 010046 MOV #MLDTSK,R3 ; Get address of loader's name
493 493 000662 CALL @SRSTD ; Search for loader's TCB
494 494 000666 103437 BCS 30$; Branch if not found
495 495 000670 010403 MOV R4,R3 ; Save system line number
496 496 000672 CALL @CCBGT ; Allocate a CCB
497 497 000676 103433 BCS 30$; Branch if allocation failed
498
499 499 000700 010401 MOV R4,R1 ; Copy CCB address
500 500 000702 062704 000006 ADD #C.LIN,R4 ; Point at system line cell
501 501 000706 010324 MOV R3,(R4)+ ; Store system line number
502 502 000710 012724 002026 MOV #FC.MLD+FS.RLB,(R4)+ ; Set function and subfunction
503 503 000714 005024 CLR (R4)+ ; Clear status
504 504 000716 012724 000006 MOV #MAXLEN,(R4)+ ; Set maximum string length
505 505 000722 011603 MOV (SP),R3 ; Recover string address
506
507 507 .IF DF M$MGE
508
509 509 000724 042703 160000 BIC #160000,R3 ; Clear APR selector in address
510 510 000730 052703 140000 BIS #140000,R3 ; Setup address for APR6 mapping
511 511 000734 MAP R2 ; Map to ASCII string
512
513 513 .ENDC
514
515 515 000740 012702 000006 MOV #MAXLEN,R2 ; Set up a loop counter
516
517 517 000744 112324 10$: MOVB (R3)+,(R4)+ ; Copy next character into CCB
518 518 000746 001402 BEQ 20$; Branch if end of string
519 519 000750 SOB R2,10$; Else, repeat
520
521 521 000754 160261 000014 20$: SUB R2,C.BUF1(R1) ; Compute correct string length
522 522 000760 CALL @EXRQF ; Queue CCB to loader and request it

```

```

152 .SBTTL PERFORM ONCE-PER-SECOND OPERATIONS
153
154 *+
155 **--ONESEC-PERFORM ONCE-PER-SECOND OPERATIONS
156 PERFORM THOSE OPERATIONS THAT ARE REQUIRED ONCE EVERY SECOND.
157
158 -
159 INPUTS:
160 R5 - FLAG FOR MULTIPROCESSOR SYSTEMS:
161 0 - GENERAL TIMER SCAN
162 <>0 - PROCESSOR SPECIFIC SCAN
163
163 000056 005777 000000G ONESEC: TST @PWRP1 ; IS POWERFAIL RECOVERY IN PROGRESS?
164 000062 100466 BMI 70$; IF M1, YES ... DON'T DISPATCH ANY TIMERS
165
166 .IF DF R$$MPL
167 .IF NDF R$$PRO
168
169 BIT #F2.MP,@FMSK2 ; IS THIS A MULTI-PROCESSOR?
170 BEQ 4$; BR IF NO
171 MOV R5,-(SP) ; SAVE DISPATCH FLAG
172 4$:
173 .ENDC
174 .ENDC
175
176 000064 017704 000000G MOV @SLTMA,R4 ; POINT TO START OF SYSTEM LINE INDEX TABLE
177 000070 017746 000000G MOV @SLTNM,-(SP) ; GET # OF LINES TO SCAN
178 000074 001433 BEQ 30$; IF EQ, NONE
179
180 .IF DF R$$MPL
181 .IF NDF R$$PRO
182
183 BIT #F2.MP,@FMSK2 ; IS THIS A MULTI-PROCESSOR?
184 BEQ 6$; BR IF NO
185 TST 2(SP) ; SHOULD WE DISPATCH LINE TIMERS?
186 BEQ 30$; IF EQ, NO
187 6$:
188 .ENDC
189 .ENDC
190
191 000076 011403 100000 MOV (R4),R3 ; GET ADDRESS OF NEXT SYSTEM LINE TABLE ENTRY
192 000100 032713 BEQ 10$; IS THIS LINE ACTIVE?
193 000104 001424 BEQ 20$; IF EQ, NO
194 000106 032713 BEQ 10$; DOES THE LINE REQUIRE TIMER SERVICE?
195 000112 001421 BEQ 20$; IF EQ, NO
196
197 .IF DF R$$MPL
198 .IF NDF R$$PRO
199
200 BIT #F2.MP,@FMSK2 ; IS THIS A MULTI-PROCESSOR?
201 BEQ 15$; BR IF NO
202 MOV L,KRBA(R3),R5 ; GET POINTER TO KRB
203 MOV @CPURM,-(SP) ; GET ADDRESS OF CPU URM TABLE
204 BIT K.URM(R5),@ (SP)+ ; IS DEVICE CONNECTED TO THIS PROCESSOR?
205 BEQ 20$; IF EQ, NO
206 15$:
207 .ENDC
208 .ENDC

```

AXTIM      CREATED BY MACRO ON 28-JUN-85 AT 18:31      PAGE 4      L 7

MACRO CROSS REFERENCE      CREF      04.00

MACRO NAME      REFERENCES

|         |        |        |        |        |        |       |        |        |        |        |
|---------|--------|--------|--------|--------|--------|-------|--------|--------|--------|--------|
| CALL    | 6-82   | 8-212  | 8-219  | 8-258  | 9-292  | 9-299 | 11-384 | 11-397 | 11-403 | 11-406 |
|         | 11-408 | 11-411 | 13-484 | 13-505 |        |       |        |        |        |        |
| CALLR   | 6-101  |        |        |        |        |       |        |        |        |        |
| CLKDF\$ | #5-57  | 5-61   |        |        |        |       |        |        |        |        |
| ENABL\$ | #5-56  | 12-455 |        |        |        |       |        |        |        |        |
| INHIB\$ | #5-56  | 12-430 |        |        |        |       |        |        |        |        |
| MTPS    | 11-347 | 11-373 | 11-416 |        |        |       |        |        |        |        |
| PDVDF\$ | #5-57  | 5-60   |        |        |        |       |        |        |        |        |
| RESRG   | #5-56  | 11-412 | 13-506 | 13-508 |        |       |        |        |        |        |
| RETURN  | 8-278  | 9-301  | 11-417 | 12-456 | 13-509 |       |        |        |        |        |
| SAVRG   | #5-56  | 11-376 | 13-467 | 13-488 |        |       |        |        |        |        |
| SLTDF\$ | #5-57  | 5-59   |        |        |        |       |        |        |        |        |
| STMDF\$ | #5-57  | 5-62   |        |        |        |       |        |        |        |        |

```

390 000746 012765 000006 000022 MOV #ERMLB,NERRC(R5)
391
392 .IF DF N$$ACC
393
394 000754 122764 000207 000010 CMPB #NT.VFY!CM.CON,C.FNC(R4)
395 000762 001407 BEQ 30$; Don't check # of links for verification requests
396
397 .ENDC
398
399 000764 105761 000011 TSTB M.MAX(R1) ; Did user restrict max # of links?
400 000770 001404 BEQ 25$; If EQ, no
401 000772 126161 000010 000011 CMPB M.USE(R1),M.MAX(R1)
402 001000 103014 BHIS 100$; If HIS, too many logical links
403 001002 25$:
404 .IF DF R$$PRO
405 CALL REMVT ; Toss the new VI: if task is already active.
406 .ENDC
407
408
409 001002 105261 000010 30$: INCB M.USE(R1) ; Update count of active links
410 001006 010167 000000G MOV R1,$MAIBX ; Save address of mailbox
411 001012 010064 000004 MOV R0,C.NSP(R4) ; Set up owning ICB address
412 001016 CALL ADDMAI ; Add mail to user's mailbox
413 001022 000402 BR 50$; Exit through common code
414
415 001024 40$: CALL ADDGNO ; Add entry to general delivery queue
416
417 001030 005727 50$: TST (PC)+ ; Indicate success
418 001032 000261 100$: SEC ; Indicate failure
419 001034 RETURN
420
421 .END

```

|     |        |        |        |       |       |           |                                         |
|-----|--------|--------|--------|-------|-------|-----------|-----------------------------------------|
| 161 | 000152 | 105264 | 000010 | 20\$: | INCB  | M,USE(R4) | ; Increment count of logical links      |
| 162 | 000156 | 010210 |        | 30\$: | MOV   | R2,(R0)   | ; Link onto the end of the mailbox list |
| 163 | 000160 | 010200 |        |       | MOV   | R2,R0     | ; Keep pointer to the end of the list   |
| 164 | 000162 | 010302 |        | 40\$: | MOV   | R3,R2     | ; Backup list pointer                   |
| 165 | 000164 | 000732 |        |       | BR    | 10\$      | ; Try for the next entry                |
| 166 |        |        |        |       |       |           |                                         |
| 167 | 000166 |        |        | 50\$: | CALLR | IOSUC     | ; Return successful I/O completion      |

```

606 .SBTTL Post event on mailbox
607
608 **--PEM--Post event on mailbox
609
610 Place an event type message on the destination tasks mailbox or
611 request the task to run.
612
613 Inputs:
614 R1 = Address of the task's header
615 R2 = Subfunction code/4
616 R3 = Address of I/O packet
617 R4 = Address of mailbox
618 R5 = Address of database descriptor
619
620 .IF DF N$$PEM!N$$ACC
621
622 000350 .PSECT $HIGH
623
624 000350 016300 000000G PEM: MOV I.TCB(R3),R0 ; Get source tasks TCB address
625 000354 016067 000000G 000000G MOV I.NAM(R0), $RQNAM; Save source task name
626 000362 016067 000002G 000002G MOV I.NAM+2(R0), $RQNAM+2
627
628 .IF DF N$$PEM&N$$ACC
629
630 000370 026727 000000G 055400 CMP $RQNAM, #*RNVP ; Is request from verification task?
631 000376 001002 BNE 5$; If NE, no
632 000400 000167 001006' JMP NVP ; Else, yes
633 000404 5$:
634 .ENDC
635
636 .IF DF N$$PEM
637
638 000404 016300 000004G 10$: MOV I.PRM+4(R3),R0 ; Get # of bytes to transfer
639 000410 027700 000000G CMP @RDBSZ,R0 ; Will it fit in a large data buffer?
640 000414 103003 BHIS 15$; If HIS, yes
641 000416 $IERRC IE.SPC&377 ; Else, illegal user buffer
642
643 000424 15$: CALL @LDBGT ; Allocate a large data buffer
644 000430 20$ BCC 20$; If CC, ok
645 000432 $IERRC IE.RSU&177 ; Else, resource allocation failure
646
647 000440 20$: SAVRG <R4>
648 000442 010064 000020 MOV R0,C.CNT(R4) ; Set up byte count in CCB
649 000446 016301 000000G MOV I.PRM(R3),R1 ; Set up source APR bias
650 000452 016302 000002G MOV I.PRM+2(R3),R2 ; Set up source virtual address
651 000456 162702 020000 SUB #20000,R2 ; for APR5
652 000462 016403 000014 MOV C.BUF(R4),R3 ; Set up destination APR bias
653 000466 016404 000016 MOV C.BUF+2(R4),R4 ; Set up destination virtual address
654 000472 CALL @BLX10 ; Copy the data
655 000476 RESRG <R4> ; Recover CCB address
656 000500 000167 000662' JMP MAPBF ; Jump to low psect
657
658 .PSECT
659
660 000662 MAPBF: MAP C.BUF(R4) ; Map to the buffer
661 000670 016402 000016 MOV C.BUF+2(R4),R2 ; Point to start of data
662 000674 010403 MOV R4,R3 ; Point to save area for

```

SESCTR - Session control counte MACRO V05.03b Friday 28-Jun-85 19:54 L 11  
Table of contents

|    |     |                               |
|----|-----|-------------------------------|
| 6- | 42  | Macro definitions             |
| 7- | 57  | Session control counter logic |
| 8- | 102 | ECL node counter routines     |
| 9- | 159 | Session control event logging |

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES |
|---------|------------|------------|
| RE\$TMR | = 000020   | #6-51      |
| RE\$UPT | = 000002   | #6-51      |
| RE\$URE | = 000003   | #6-51      |
| RE\$VER | = 000005   | #6-51      |
| RE\$VRQ | = 000015   | #6-51      |
| RT\$INI | = 000002   | #6-51      |
| RT\$OFF | = 000001   | #6-51      |
| RT\$ON  | = 000000   | #6-51      |
| SC\$OFF | = 000001   | #6-51      |
| SC\$ON  | = 000000   | #6-51      |
| SC\$RST | = 000003   | #6-51      |
| SC\$SHU | = 000002   | #6-51      |
| SV\$DUM | = 000001   | #6-51      |
| SV\$LQA | = 000000   | #6-51      |
| T\$FLAG | 000044     | #6-48      |
| T\$LIF  | 000013     | #6-48      |
| T\$LIFL | 000013     | #6-48      |
| T\$LIFO | 000013     | #6-48      |
| T\$LIFS | 000013     | #6-48      |
| T\$LIN  | 000000     | #6-48      |
| T\$LIPS | 000006     | #6-48      |
| T\$LLD  | 000012     | #6-48      |
| T\$LLDC | 000045     | #6-48      |
| T\$LLDL | 000012     | #6-48      |
| T\$LLDO | 000012     | #6-48      |
| T\$LLDS | 000012     | #6-48      |
| T\$LLEN | 000046     | #6-48      |
| T\$LOPR | 000002     | #6-48      |
| T\$LTCL | 000024     | #6-48      |
| T\$LTIM | 000026     | #6-48      |
| T\$LTPR | 000014     | #6-48      |
| T\$LTPS | 000020     | #6-48      |
| T\$NAPL | 000004     | #6-48      |
| T\$NFE  | 000000     | #6-48      |
| T\$NLEN | 000010     | #6-48      |
| T\$NNUL | 000002     | #6-48      |
| T\$NOPL | 000006     | #6-48      |
| T\$NRNI | 000042     | #6-48      |
| T\$NRPL | 000005     | #6-48      |
| T\$NRUL | 000007     | #6-48      |
| T\$NVR  | 000001     | #6-48      |
| T\$RPRI | 000040     | #6-48      |
| T\$SVC  | 000034     | #6-48      |
| T\$T5   | 000030     | #6-48      |
| T\$T6   | 000032     | #6-48      |
| BYTE    | = ***** GX | 8-137      |



## SYMBOL CROSS REFERENCE

CREF 04.00

SYMBOL VALUE REFERENCES

```

X$$HDR = ***** 7-93
ZTIM2 000366 RG #9-188
$ACNT 000202 RG #8-124
$ALOCB = ***** GX 9-135
$BLXIO = ***** GX 9-136
$BYTE 000020 RG #7-62
$CALLX = ***** GX 9-159
$CCBGT = ***** GX 9-160
$CCBRT = ***** GX 9-161
$CEACC = ***** GX 9-162
$CEDIV = ***** GX 9-166
$CELOG = ***** GX 9-163
$CEMUL = ***** GX 9-167
$CMPDV = ***** GX 9-170
$CNBLK 000070 RG #8-113
$CSBGT = ***** GX 9-171
$CSBRT = ***** GX 9-172
$DEACB = ***** GX 9-137
$DECPT = ***** GX 9-173
$DEVHD = ***** GX 9-138
$DREXT = ***** GX 9-139
$DSDFM 000076 RG #8-115
$DSDSC 000100 RG #8-118
$DSNOD 000070 RG #8-114
$DSOBJ 000077 RG #8-116
$ENCOD 000036 RG #7-73
$EVDSC = ***** GX 9-174
$FLAGS 000047 RG #7-87
$FLOW 000046 RG #7-79
$FMASK = ***** GX 9-140
$INFO 000025 RG #7-66
$IOFIN = ***** GX 9-141
$IOPKT 000042 RG #7-76
$LADDR 000032 RG #7-69
$LDBGT = ***** GX 9-175
$LDBRT = ***** GX 9-176
$LLCRS = ***** GX 9-177
$LTM 000044 RG #7-78
$MAIBX 000016 RG #7-60
$MENU 000030 RG #7-68
$MVFBF = ***** GX 9-178
$NMCLH = ***** GX 9-179
$OBJHD = ***** GX 9-180
$OPDAT 000226 RG #8-126
$OPLNG 000224 RG #8-125
$PASSW 000170 RG #8-123
$PDVID = ***** GX 9-181
$PDVTA = ***** GX 9-182
$QRMVF = ***** GX 9-142
$QUEBF = ***** GX 9-143
$RCCB 000022 RG #7-64
$RDBRT = ***** GX 9-183

```

\*\*\*FILE\*\*ID\*\*SESDMO

L 14

```
SSSSSSSS EEEEEEEEE SSSSSSS DDDDDDD MM MM 000000
SSSSSSSS EEEEEEEEE SSSSSSS DDDDDDD MM MM 000000
SS EE SS SS DD DD MMMM MMMM 00 00
SS EE SS SS DD DD MMMM MMMM 00 00
SS EE SS SS DD DD MM MM MM 00 00
SS EE SS SS DD DD MM MM MM 00 00
SSSSSS EEEEEEEEE SSSSSS DD DD MM MM 00 00
SSSSSS EEEEEEEEE SSSSSS DD DD MM MM 00 00
SS EE SS SS DD DD MM MM 00 00
SS EE SS SS DD DD MM MM 00 00
SS EE SS SS DD DD MM MM 00 00
SSSSSS EEEEEEEEE SSSSSSS DDDDDDD MM MM 000000
SSSSSS EEEEEEEEE SSSSSSS DDDDDDD MM MM 000000
.....
.....
.....
.....
```

```
11 11 SSSSSSS
11 11 SSSSSSS
1111 1111 SS
1111 1111 SS
11 11 SS
11 11 SS
11 11 SSSSS
11 11 SSSSS
11 11 SS
11 11 SS
11 11 SS
11 11 SS
111111 111111 SSSSSSS
111111 111111 SSSSSSS
```

SESDM011S CREATED BY MACRO ON 28-JUN-85 AT 19:55

PAGE 6 L 15

MACRO CROSS REFERENCE

CREF 04.00

MACRO NAME REFERENCES

|         |        |       |       |
|---------|--------|-------|-------|
| CALL    | 7-91   | 7-99  | 8-131 |
| CALLE   | #6-44  | 8-126 |       |
| CALLR   | 7-104  |       |       |
| CALLX   | #8-126 | 8-126 |       |
| ECDDBS  | #6-45  | 6-47  |       |
| EVLDF\$ | #6-45  | 6-48  |       |
| EVT\$   | #6-44  | 7-74  |       |
| LLTDF\$ | #6-45  | 6-49  |       |
| MAP     | #6-44  |       |       |
| MAPLLT  | #6-44  |       |       |
| MSGDF\$ | #6-45  | 6-50  |       |
| RESRG   | #6-44  | 8-132 |       |
| RETURN  | 7-106  | 8-135 |       |
| SAVRG   | #6-44  | 8-128 |       |
| SOB     | 8-134  |       |       |

```

393 .ENDC
394
395 .IF DF N$MCP
396
397 10$: BITB #CX.SMC,C.MOD(R4) ; Path for tasks installed in primary pool
398 BEQ 30$; Spawn multiple copies?
399 ; If EQ, no - use this TCB
400
401 CMP T.NAM+2(P0),#^R$$$; Else, check for valid installed name
402 BNE 30$; If NE, can't do a multi copy from it
403
404 20$: CALL MPICB ; Create a new tcb for this task
405 BCS 60$; If CS, allocation failure
406
407 30$: .ENDC
408
409 000354 005001 CLR R1 ; Assume default UIC
410 000356 132764 000040 000011 BITB #CX.RUI,C.MOD(R4)
411 000364 001404 BEQ 40$; If eq, use default UIC
412 000366 016401 000016 MOV C.BUF+2(R4),R1 ; Get login UIC from connect request block
413 000372 016101 000066 MOV N.CUIC(R1),R1 ; ...
414
415 000376 40$: SAVRG <R0> ; Save TCB address
416
417 .IF DF R$PRO
418 MOV C.PRO(R4),R2 ; Get the VT: UCB address.
419 CALL $TSKRQ ; Activate the task under the VT:
420 .IFF ; DF R$PRO
421
422 000400 CALL @TSKRT ; Request task to execute
423 .IFTF ; DF R$PRO
424
425 000404 RESRG <R0> ; Recover TCB address
426 000406 103002 BCC 45$; If CC, requested task successfully
427 000410 001013 BNE 50$; If CS, and <>0, task allocation failure
428
429 .IFT ; DF R$PRO
430 RECMAP ; Recover NETACP's KISAR6
431 CALL REMVT ; If the task is already active, dissolve
432 ; ...the new virtual terminal.
433 .ENDC ; DF R$PRO
434
435 000412 000241 CLC ; If the task is already active, indicate
436 ; ...success.
437 000414 142764 000002 000011 45$: BICB #CX.REQ,C.MOD(R4)
438 000422 132764 000020 000011 BITB #CX.REM,C.MOD(R4)
439 000430 001403 BEQ 50$; If EQ, we did not generate a new TCB
440 000432 052760 000000G 000000G BIS #T3.REM,T.ST3(R0)
441
442 000440 010064 000004 50$: MOV R0,C.NSP(R4) ; Save TCB address
443 000444 60$: RECMAP ; Recover high APR mapping
444 000452 RETURN
445
446 ;+
447 ; Dispatch table for task name determination
448 ;+
449 000352 .PSECT $HIGH

```

```

166 .SBTTL LOOK FOR RING
167
168 ;*-RING-LOOK FOR RING
169
170 ;
171 ; WAIT UNTIL A RING TONE HAS BEEN DETECTED INDICATING THAT SOMEONE
172 ; IS TRYING TO DIAL IN.
173
174 ;
175 ; INPUTS:
176 ; R5 - POINTER TO LINE ENTRY IN MDC DATABASE
177 ;
178 ; RING: TSTB M.STT(R5) ; IS THE LINE NOW IDLE?
179 ; BNE 10$; NO ... IGNORE THIS REQUEST
180 ; MOVB #ST.WRG,M.STT(R5) ; SET NEW STATE - WAIT FOR RING
181 ; CLRB M.PSV(R5) ; INITIALIZE SERVICE BITS
182 ; CLRB M.CSV(R5) ; ...
183 ;
184 ; 10$: RETURN
185 ;

```

```

568 .SBTTL ST.ADL - ACTIVE DELAY
569
570
571 ***.ACTD-ACTIVE DELAY
572
573 LOSS OF CARRIER WAS DETECTED ON AN ASYNCHRONOUS LINE, DELAY AND
574 RECHECK IN CASE THE LOSS WAS ONLY TEMPORARY.
575
576 INPUTS:
577 R2 - POINTER TO SYSTEM LINE TABLE
578 R4 - SYSTEM LINE NUMBER
579 R5 - POINTER TO LINE ENTRY IN MDC DATABASE
580 000720 132765 000001 000004 .ACTD: BITB #MC.CAR,M.CSV(R5) ; HAS CARRIER RETURNED?
581 000726 001005 BNE 10$; YES ... RETURN TO ACTIVE
582 000730 105765 000002 TSTB M.TIM(R5) ; NO ... HAS THE TIMER EXPIRED?
583 000734 001007 BNE 20$; NO ... KEEP WAITING
584 000736 CALLR DISC ; POST ASYNCHRONOUS DISCONNECT
585 000742 105065 000002 10$: CLRB M.TIM(R5) ; CARRIER HAS RETURNED ... CANCEL TIMER
586 000746 112765 000010 000003 MOVB #ST.ACT,M.STT(R5) ; RETURN TO ACTIVE STATE
587 000754 20$: RETURN
588

```

\*\*FILE\*\*ID\*\*AXSCH

```

AAAAAA XX XX SSSSSSSS CCCCCCCC HH HH
AAAAAA XX XX SSSSSSSS CCCCCCCC HH HH
AA AA XX XX SS CC HH HH
AA AA XX XX SS CC HH HH
AA AA XX XX SS CC HH HH
AA AA XX XX SS CC HH HH
AA AA XX XX SS CC HH HH
AAAAAA XX XX SSSSSS CC HHHHHHHHHH
AAAAAA XX XX SSSSSS CC HHHHHHHHHH
AA AA XX XX SS CC HH HH
AA AA XX XX SS CC HH HH
AA AA XX XX SS CC HH HH
AA AA XX XX SSSSSSSS CCCCCCCC HH HH
AA AA XX XX SSSSSSSS CCCCCCCC HH HH

```

```

....
....
....
....

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```

|     |     |                                                   |
|-----|-----|---------------------------------------------------|
| 5-  | 51  | Macro definitions                                 |
| 8-  | 172 | \$ALOCX - Allocate core buffer in extended pool   |
| 9-  | 242 | \$DEACX - Deallocate core buffer in extended pool |
| 10- | 342 | \$XLINK - Link/unlink block in extended pool      |
| 11- | 422 | \$CMINI - Component buffer limit (init)           |
| 12- | 466 | \$RQSTL - Request System Microcode Loader         |



```

523 000764 000241 CLC ; OK if task already fixed
524
525 000766 012603 30$: MOV (SP)+,R3 ; Recover string address
526 000770 012601 MOV (SP)+,R1 ; Recover a register
527 000772 012600 MOV (SP)+,R0 ; Recover a register
528 000774 012703 100100 MOV #CS.ERR+CS.AB0,R3 ; Assume task request failed
529 001000 103401 BCS 40$; Branch if it really did fail
530 001002 005003 CLR R3 ; Else, return success status
531
532 001004 40$: RETURN ; Exit
533
534 000001 .END

```

```

209
210 000114 016305 000004 MOV L.DDS(R3),R5 ; GET DDM LINE TABLE ADDRESS
211 000120 116302 000002 MOV L.DDM(R3),R2 ; AND DDM PDV INDEX
212 000124 CALL @TSTIM ; TEST TIMEOUT AND DISPATCH IF REQUIRED
213
214 000130 011403 MOV (R4),R3 ; RECOVER SYSTEM LINE TABLE ENTRY ADDRESS
215 000132 126363 000002 000003 CMPB L.DDM(R3),L.DLC(R3) ; COMBINED DLC/DDM?
216 000140 001406 BEQ 20$; IF EQ, YES
217 000142 016305 000010 MOV L.DLS(R3),R5 ; GET DLC LINE TABLE ADDRESS
218 000146 116302 000003 MOV L.DLC(R3),R2 ; AND DLC PDV INDEX
219 000152 CALL @TSTIM ; TEST TIMEOUT AND DISPATCH IF REQUIRED
220
221 000156 20$: .IF NDF N$$1LN
222
223 000156 005724 TST (R4)+ ; MOVE ON TO NEXT ENTRY
224 000160 005316 DEC (SP) ; REDUCE SYSTEM LINE COUNT
225 000162 001345 BNE 10$; AND LOOP TILL ALL DONE
226
227 .ENDC
228
229 000164 017704 000000G 30$: MOV @PDVTA,R4 ; GET ADDRESS OF PDV INDEX TABLE
230 000170 017716 000000G MOV @PDVNM,(SP) ; AND # OF PDV'S TO SCAN
231
232 000174 012401 40$: MOV (R4)+,R1 ; GET NEXT PDV ADDRESS
233 000176 005761 000012 TST Z.PCB(R1) ; IS THE PROCESS LOADED?
234 000202 001414 BEQ 60$; IF EQ, NO
235 000204 032761 000200 000010 BIT #ZF.TIM,Z.FLG(R1) ; DOES THIS PROCESS REQUIRE TIMER SERVICE?
236 000212 001410 BEQ 60$; IF EQ, NO
237
238 .IF DF R$$MPL
239 .IF NDF R$$PRO
240
241 BIT #F2.MP,@FMSK2 ; IS THIS A MULTI-PROCESSOR?
242 BFO 50$; BR IF NO
243 TST 2(SP) ; PROCESSOR DEPENDANT SCAN?
244 BEQ 50$; IF EQ, NO
245 BIT #ZF.MTM,Z.FLG(R1) ; DOES PROCESS REQUIRE MULTIPLE TIMERS?
246 BEQ 60$; IF EQ, NO
247 MOV @CPURM,-(SP) ; GET ADDRESS OF CPU URM TABLE
248 BIT @CKURM,@(SP)+ ; DOES THIS PROCESSOR CONTROL THE CLOCK?
249 BNE 60$; IF NE, YES ... WE HAVE ALREADY DONE THE SCAN
250
251 50$: .ENDC
252 .ENDC
253
254 0002 4 016105 000016 MOV Z.DAT(R1),R5 ; GET ADDRESS OF DATABASE DESCRIPTOR
255 000220 010402 MOV R4,R2 ; COMPUTE PDV INDEX
256 000222 167702 000000G SUB @PDVTA,R2 ; ...
257 000226 005742 TST -(R2) ; ...
258 000230 CALL @DSPTM ; DISPATCH PROCESS TIMEOUT
259
260 000234 005316 60$: DEC (SP) ; REDUCE PDV COUNT
261 000236 001356 BNE 40$; AND LOOP TILL ALL DONE
262
263 000240 70$: .IF NDF R$$PRO
264 .IF DF R$$MPL
265

```

```

NNN NNN EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT 111 111 SSSSSSSSSSS
NNN NNN EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT 111 111 SSSSSSSSSSS
NNN NNN EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT 111 111 SSSSSSSSSSS
NNN NNN EEE TTT 111111 111111 SSS
NNN NNN EEE TTT 111111 111111 SSS
NNN NNN EEE TTT 111111 111111 SSS
NNNNNNN NNN EEE TTT 111 111 SSS
NNNNNNN NNN EEE TTT 111 111 SSS
NNNNNNN NNN EEE TTT 111 111 SSS
NNNNNNN NNN EEE TTT 111 111 SSS
NNN NNN EEEEEEEEEEE TTT 111 111 SSSSSSSSS
NNN NNN EEEEEEEEEEE TTT 111 111 SSSSSSSSS
NNN NNN EEEEEEEEEEE TTT 111 111 SSSSSSSSS
NNN NNNNNN EEE TTT 111 111 SSS
NNN NNNNNN EEE TTT 111 111 SSS
NNN NNNNNN EEE TTT 111 111 SSS
NNN NNN EEE TTT 111 111 SSS
NNN NNN EEE TTT 111 111 SSS
NNN NNN EEE TTT 111 111 SSS
NNN NNN EEEEEEEEEEEEEEE TTT 111111111 111111111 SSSSSSSSSSS
NNN NNN EEEEEEEEEEEEEEE TTT 111111111 111111111 SSSSSSSSSSS
NNN NNN EEEEEEEEEEEEEEE TTT 111111111 111111111 SSSSSSSSSSS

```

|                  |                 |                  |                 |                  |
|------------------|-----------------|------------------|-----------------|------------------|
| ACC 000300R      | CP.PSI= 000200  | C.NSP 000004     | ER\$ABM= 000010 | FS.STP= 002000   |
| ADDGNQ= ***** GX | CP.XCF= 000100  | C.PRO 000042     | ER\$ABO= 000046 | FS.STR= 001000   |
| ADDMAI= ***** GX | CP.2FR= 000030  | C.RSV 000002     | ER\$ABT= 000011 | FS.TRM= 003000   |
| AS\$CHK= 000000  | CS.ABO= 000100  | C.STA 000007     | ER\$ACC= 000042 | FS.WLB= 001000   |
| AS\$CPS= 000000  | CS.BRO= 000002  | C.STS 000012     | ER\$CDI= 000052 | FS.XKL= 002000   |
| AS\$PRI= 000000  | CS.BUF= 000200  | C.URM 177776     | ER\$COM= 000047 | FS.XOF= 010000   |
| AS\$TR= 000000   | CS.CES= 000002  | C.XACP 000004    | ER\$FMT= 000005 | FS.XOM= 007000   |
| CAT5 = ***** GX  | CS.CHN= 000010  | C.XID 000035     | ER\$MLB= 000006 | FS.ZER= 002000   |
| CB.CCB= 000002   | CS.CMP= 000200  | C.XLEN 000044    | ER\$NNF= 000012 | F\$SLVL= 000001  |
| CB.DDM= 000040   | CS.DCR= 000400  | C.XPLI 000040    | ER\$NOD= 000002 | G\$STPP= 000000  |
| CB.DLC= 000020   | CS.DEF= 000004  | C.XPT 000034     | ER\$NSL= 000013 | G\$STSS= 000000  |
| CB.RDB= 000004   | CS.DEV= 000002  | C.XSVC 000042    | ER\$NSR= 000003 | G\$STTK= 000000  |
| CB.SDB= 000010   | CS.DIS= 000040  | C.XTC 000037     | ER\$RES= 000001 | G\$SWRD= 000000  |
| CB.SLI= 000100   | CS.ENA= 000001  | C.X25 000036     | ER\$STA= 000051 | IE.BAD= ***** GX |
| CB.XLB= 000001   | CS.ENB= 000020  | CSTA = ***** GX  | ER\$UNP= 000004 | IE.NRJ= ***** GX |
| CC.LLC= 000200   | CS.ERR= 100000  | DECTP = ***** GX | E\$X-R= 000000  | IE.RSU= ***** GX |
| CE.ABO= 100362   | CS.FTL= 001000  | D\$AMXC 000072   | FC.CCP= 000020  | IN.DAT= 000400   |
| CE.DAO= 100346   | CS.HCR= 000001  | D\$AMXH 000074   | FC.CTL= 000006  | IN.ILS= 000001   |
| CE.DIS= 100366   | CS.HFE= 002000  | D\$ANN 000000    | FC.KCP= 000016  | IODUN = ***** GX |
| CE.ERR= 100370   | CS.LST= 040000  | D\$BRPR 000102   | FC.KIL= 000004  | IOERR = ***** GX |
| CE.ILN= 100350   | CS.MTL= 004000  | D\$BRTM 000100   | FC.MAN= 000024  | IOSUC = ***** GX |
| CE.LTD= 100356   | CS.RNG= 000010  | D\$DELF 000045   | FC.MLD= 000026  | I\$SRAR= 000000  |
| CE.MOP= 100372   | CS.ROV= 000004  | D\$DELF 000046   | FC.PCT= 000030  | I\$SRDN= 000000  |
| CE.NTE= 100361   | CS.RSN= 010000  | D\$END = 000104  | FC.PWR= 000022  | I.FCN = ***** GX |
| CE.RTE= 100376   | CS.SHU= 000001  | D\$FNB 000034    | FC.RCE= 000002  | I.PRM = ***** GX |
| CE.SRC= 100364   | CS.SID= 000002  | D\$HIOR 000024   | FC.RCP= 000014  | I.TCB = ***** GX |
| CE.STP= 100352   | CS.STR= 000004  | D\$HOST 000022   | FC.TIM= 000010  | KISAR6= ***** GX |
| CE.TME= 100354   | CS.SUC= 000001  | D\$INAC 000044   | FC.XCP= 000012  | K\$SCNT= 177546  |
| CE.TMO= 100374   | CS.TMO= 020000  | D\$INCT 000042   | FC.XME= 000000  | K\$SCSR= 177546  |
| CE.UNS= 100344   | CS.XUR= 000004  | D\$IPL 000051    | FS.AST= 000000  | K\$SLDC= 000000  |
| CF.CHN= 000001   | CV\$MSK= 000003 | D\$LID 000020    | FS.CIB= 002000  | K\$STPS= 000074  |
| CF.EOM= 00'304   | CV\$31 = 000001 | D\$LNAM 000006   | FS.CRA= 001000  | LA.ACK= 100000   |
| CF.HDR= 000020   | CV\$32 = 000000 | D\$LNAM 000014   | FS.DIS= 013000  | LA.CRS= 020000   |
| CF.LB = 100000   | CV\$40 = 000002 | D\$LST 000047    | FS.DVC= 001000  | LA.MSK= 170000   |
| CF.LIN= 000002   | CX.GD= 000001   | D\$MAXC 000064   | FS.ENB= 012000  | LA.NAK= 110000   |
| CF.SOM= 000010   | CX.REM= 000020  | D\$MAXH 000066   | FS.EXI= 001000  | LA.NMS= 010000   |
| CF.SYN= 000040   | CX.REQ= 000002  | D\$MAXV 000070   | FS.GET= 006000  | LA.RES= 040000   |
| CF.TRN= 000100   | CX.RUI= 000040  | D\$MLL 000040    | FS.HLT= 000000  | LA.WND= 004000   |
| CL\$MFL= 000010  | CX.SMC= 000010  | D\$MNOD 000041   | FS.INI= 000000  | LD\$LP = 000000  |
| CL\$SFL= 000004  | CX.UNL= 000004  | D\$NA 000062     | FS.KIL= 000000  | LF.DRD= 000004   |
| CL\$TYP= 000001  | C\$SORE= 000400 | D\$NBEA 000036   | FS.LCL= 100000  | LF.FRC= 000001   |
| CL.MU1= 000001   | C\$SRSH= 177564 | D\$NBRA 000034   | FS.LTM= 001000  | LF.HFO= 000010   |
| CL.MU2= 000002   | C.ADD 000034    | D\$NEND= 000034  | FS.MNT= 004000  | LF.HMK= 000040   |
| CL.RES= 177774   | C.BID 000003    | D\$NLN 000030    | FS.MSN= 014000  | LF.HSF= 000020   |
| CM.CIR= 000002   | C.BUF 000014    | D\$NN 000060     | FS.REA= 001000  | LF.IRD= 000002   |
| CM.CON= 000200   | C.BUF1 000014   | D\$OUTT 000043   | FS.RET= 000000  | LF.MMF= 000200   |
| CM.FMT= 100000   | C.BUF2 000024   | D\$RETF 000050   | FS.REZ= 003000  | LF.MSF= 000100   |
| CM.HRD= 000002   | C.CNT 000020    | D\$RNN 000002    | FS.RLB= 002000  | LS.DLS= 100000   |
| CM.LIN= 000000   | C.CNT1 000020   | D\$RTMR 000076   | FS.RNG= 011000  | LS.FCC= 000004   |
| CM.LOD= 000001   | C.CNT2 000030   | D\$SEG 000036    | FS.RST= 000000  | LS.FCO= 000001   |
| CM.XLO= 000004   | C.FLG 000022    | D\$SER 000032    | FS.RTN= 001000  | LS.FCI= 000002   |
| CON 000106R      | C.FLG1 000022   | D\$SGRL 000032   | FS.SET= 005000  | LS.ILS= 100000   |
| CONBL 000000R    | C.FLG2 000032   | D\$BUG= 177514   | FS.SFC= 005000  | LS.MAK= 000020   |
| CPYOPT= ***** GX | C.FNC 000010    | D\$ISK= 000000   | FS.SFR= 006000  | LS.MNK= 000040   |
| CP.DCF= 000040   | C.LIN 000006    | D\$SL11= 000001  | FS.SFS= 004000  | LS.RES= 000360   |
| CP.HDL= 000007   | C.LNK 000000    | D\$SYNC= 000000  | FS.SPW= 040000  | LS.RSV= 000300   |
| CP.PS = 177400   | C.MOD 000011    | D\$SYNM= 000000  | FS.STM= 000000  | LT.CCA= 000020   |

```

169 .SBTTL Network deaccess
170
171 ;+
172 ;**-$CLOIO-Close LUN request
173 ;**-$CLS-Network deaccess
174
175 This routine cleans up the mailbox and any logical links the user
176 has active.
177
178 Inputs:
179 R1 = Address of the task's header
180 R2 = Subfunction code/4
181 R3 = Address of I/O packet
182 R4 = Address of mailbox
183 R5 = Address of database descriptor
184
185 .PSECT
186
187 $CLOIO::
188 CLS: MOV R3,M.SPA(R4) ; Save address of I/O packet
189 SAVRG <R1> ; Save address of task header
190
191 MOV R4,R3 ; Copy address of mailbox
192 CLR R0 ; Set flag to flush all events from mailbox
193 CALL FLSHMB ; Flush the mailbox
194
195 TSTB M.USE(R4) ; Are any logical links active?
196 BEQ CLSDON ; If EQ, no
197
198 MOV (SP),R1 ; Recover task header address
199
200 .IF DF X$$HDR
201 MAP $HDRMP ; Map to the task's header
202
203 .ENDC
204
205 MOV H.NLUN(R1),-(SP) ; Get # of LUNs to scan
206 ADD #H.LUN,R1 ; Point to first LUN
207
208 10$: .IF DF X$$HDR
209 MAP $HDRMP ; Map to the task's header
210
211 .ENDC
212
213
214 CMP (R1)+,$UCB ; Is LUN assigned to us?
215 BNE 100$; If NE, no
216 MOV (R1),R3 ; Get window block address
217 BIC #1,R3 ; Remove possible LUN interlock
218 BEQ 100$; If EQ, no active link
219 CMP R3,R4 ; Is this the mailbox LUN?
220 BEQ 100$; If EQ, yes ... can't disconnect it
221
222 SAVRG <R1,R3,R4> ; Save some registers
223 MOV W.LLT(R3),-(SP) ; Save address of LLT
224 MOV R3,R4 ; Copy window block address
225 CALL FLSHIO ; Flush any outstanding I/O packets

```

```

663 000676 062703 000030 ADD #C.CNT2,R3 ; destination task name
664 000702 012213 MOV (R2)+(R3) ; Save destination task name
665 000704 012263 000002 MOV (R2)+,2(R3) ; ...
666
667 000710 016742 000002G MOV $RQNAM+2,-(R2) ; Move source task name into the buffer
668 000714 016742 000000G MOV $RQNAM,-(R2) ; ...
669
670 .IF DF R$$PRO
671 MOV $COPT,C.PRO(R4) ; Store UCB address of CO:
672 .ENDC ; DF R$$PRO
673
674 .IF DF R$$MPL
675 CALL @SRPRO ; Search for TCB address
676 .IFF ; DF R$$MPL
677
678 000720 CALL @SRSTD ; Search for TCB address
679 .ENDC ; DF R$$MPL
680
681 000724 103423 BCS 100$; If CS, task not installed
682 000726 RECMAP ; Recover ACP mapping
683 000734 012764 004006 000010 MOV #<CX.SMC+400>+NT.EVT,C.FNL(R4)
684 000742 112764 000000G 000013 MOVB #N$$SMC,C.STS+1(R4)
685 000750 CALL FNDMBX ; Try to find task's mailbox
686 000754 103003 BCC 20$; If CC, found it
687
688 000756 CALL ADDGNO ; Add CCB to general delivery queue
689 000762 000402 BR 30$; and complete the I/O
690
691 000764 20$: CALL ADDMAI ; Add mail to mailbox
692 000770 30$: CALLR IOSUC ; Complete the I/O request successfully
693
694 000774 100$: CALL @LDBRT ; Return the buffer
695 001000 $IERRC IE.INS&377 ; Task not installed
696
697
698 .ENDC

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

.TITLE SESCTR - Session control counter logic and event logging  
.IDENT /V05.00/  
.ENABL LC

Copyright (C) 1982, 1983, 1985 by  
Digital Equipment Corporation, Maynard, MASS.

This software is furnished under a license for use only on a  
single computer system and may be copied only with the  
inclusion of the above copyright notice. This software, or  
any other copies thereof, may not be provided or otherwise  
made available to any other person except for use on such  
system and to one who agrees to these license terms. Title  
to and ownership of the software shall at all times remain  
in DEC.

The information in this document is subject to change without  
notice and should not be construed as a commitment by Digital  
Equipment Corporation.

DEC assumes no responsibility for the use or reliability of  
its software on equipment which is not supplied by DEC.

#### Module description

Session control counter logic and event logging

#### Ident history:

- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RX V1.0

SESCTR11S CREATED BY MACRO ON 28-JUN-85 AT 19:54 PAGE 6 M 12

MACRO CROSS REFERENCE

CREF 04.00

MACRO NAME REFERENCES

|         |       |       |       |       |
|---------|-------|-------|-------|-------|
| CALL    | 7-85  | 9-192 |       |       |
| CCBDF\$ | #6-45 | 6-47  |       |       |
| CTRDF\$ | #6-45 | 6-48  |       |       |
| ECDDB\$ | #6-45 | 6-49  |       |       |
| EVLD\$  | #6-45 | 6-51  |       |       |
| LLTDF\$ | #6-45 | 6-50  |       |       |
| MAP     | #6-44 | 7-74  | 7-81  |       |
| MAPLLT  | #6-44 | 7-74  |       |       |
| RESMAP  | #6-44 | 7-87  |       |       |
| RESRG   | #6-44 | 7-90  | 9-193 | 9-195 |
| RETURN  | 8-124 | 8-142 | 8-151 |       |
| SAVMAP  | #6-44 | 7-73  |       |       |
| SAVRG   | #6-44 | 7-72  | 9-176 | 9-191 |



SESDAT11S CREATED BY MACRO ON 28-JUN-85 AT 19:55      PAGE 3      M 13  
 SYMBOL CROSS REFERENCE      CREF 04.00

| SYMBOL  | VALUE      | REFERENCES |
|---------|------------|------------|
| \$RDBSZ | = ***** GX | 9-184      |
| \$REASN | 000034 RG  | #7-71      |
| \$REQID | 000146 RG  | #8-122     |
| \$RQCPY | 000014 RG  | #7-58      |
| \$RQNAM | 000010 RG  | #7-56      |
| \$RQTCB | 000006 RG  | #7-55      |
| \$SEGMT | 000026 RG  | #7-67      |
| \$SESDB | 000000 RG  | #7-50      |
| \$SESPD | 000002 RG  | #7-51      |
| \$SRDFM | 000122 RG  | #8-119     |
| \$SRDSC | 000124 RG  | #8-121     |
| \$SROBJ | 000123 RG  | #8-120     |
| \$SRSTD | = ***** GX | 9-149      |
| \$SRVCS | 000024 RG  | #7-65      |
| \$STPCT | = ***** GX | 9-150      |
| \$STKCB | = ***** GX | 9-156      |
| \$TSKRT | = *****    | 9-157      |
| \$TTNS  | = ***** GA | 9-158      |
| \$UCB   | 000004 RG  | #7-53      |
| \$VECLN | = 000051 G | #9-190     |
| \$VECTB | 000246 RG  | #9-130     |
| \$WBLK  | 000040 RG  | #7-75      |
| \$WORK  | 000100 RG  | #8-117     |
| \$ZTIM2 | = ***** GX | 9-188      |

SESDMO - Session control dismount MACRO V05.03b Friday 28-Jun-85 19:55 <sup>N 14</sup>  
Table of contents

6- 42 Macro definitions  
7- 54 Dismount QIO processing  
8- 108 Flush any active logical links

••FILE••ID••SESDSP

M 15

```
SSSSSSSS EEEEEEEEE SSSSSSSS DDDDDDDD SSSSSSSS PPPPPPPP
SSSSSSSS EEEEEEEEE SSSSSSSS DDDDDDDD SSSSSSSS PPPPPPPP
SS EEE SS DD DD SS PP PP
SS EEE SS DD DD SS PP PP
SS EEE SS DD DD SS PP PP
SSSSSS EEEEEEEEE SSSSSS DD DD SSSSSS PPPPPPPP
SSSSSS EEEEEEEEE SSSSSS DD DD SSSSSS PPPPPPPP
 SS SS DD DD SS PP
 SS SS DD DD SS PP
 SS SS DD DD SS PP
SSSSSSSS EEEEEEEEE SSSSSSSS DDDDDDDD SSSSSSSS PP
SSSSSSSS EEEEEEEEE SSSSSSSS DDDDDDDD SSSSSSSS PP
SSSSSSSS EEEEEEEEE SSSSSSSS DDDDDDDD SSSSSSSS PP
```

```
11 11 SSSSSSSS
11 11 SSSSSSSS
1111 1111 SS
1111 1111 SS
11 11 SS
11 11 SS
11 11 SSSSSS
11 11 SSSSSS
11 11 SS
11 11 SS
11 11 SS
11 11 SS
111111 111111 SSSSSSSS
111111 111111 SSSSSSSS
```

M 16

SESDSP - Session control dispat MACRO V05.03b Friday 28-Jun-85 19:56 Page 11-2  
 Request task to run

|     |        |         |              |                                              |
|-----|--------|---------|--------------|----------------------------------------------|
| 450 |        |         |              |                                              |
| 451 | 000352 | 000370' | NDISP: .WORD | NMCON ; Connect request                      |
| 452 | 000354 | 000355' | .WORD        | .+1 ; Interrupt message                      |
| 453 | 000356 | 000357' | .WORD        | .+1 ; Disconnect                             |
| 454 | 000360 | 000361' | .WORD        | .+1 ; comes in 3 flavours                    |
| 455 | 000362 | 000363' | .WORD        | .+1                                          |
| 456 | 000364 | 000376' | .WORD        | NMEVT ; Network event                        |
| 457 |        |         |              |                                              |
| 458 |        |         | .IF DF       | N\$\$\$ACC                                   |
| 459 | 000366 | 000406' | .WORD        | NMVFY ; Verification request                 |
| 460 |        |         | .IFF         |                                              |
| 461 |        |         | .WORD        | .+1 ; Verification request (never generated) |
| 462 |        |         | .ENDC        |                                              |
| 463 |        |         |              |                                              |
| 464 |        |         | ; .WORD      | NMMOP ; Mop event                            |

```

184 .SBTTL ENABLE LINE
185
186 :+
187 : **--ENABLE-ENABLE LINE
188 :
189 : ENABLE THE LINE BY WAITING FOR CARRIER OR DATASET READY TO BE
190 : ASSERTED.
191 :
192 : INPUTS:
193 : R5 - POINTER TO LINE ENTRY IN MDC DATABASE
194
195 000114
196
197 :
198 : TSTB M.STT(R5) ; IS THE LINE CURRENTLY IDLE?
199 : BNE 10$; NO ... IGNORE THIS REQUEST
200
201 000114 112765 000004 000003 MOVB #ST.WCN, M.STT(R5) ; YES ... SET STATE - WAIT FOR CONNECT
202 000122 105065 000005 CLR B M.PSV(R5) ; INITIALIZE SERVICE BITS
203 000126 105065 000004 CLPB M.CSV(R5) ; ...
204
205 : IF DF R$SP
206 : IF NDF X$SD
207 BICB #MS.SYN, M.STS(R5) ; ASSUME CARRIER AND DSR TO BE MONITORED
208 MOV @SPAR1, -(SP) ; ADDRESS OF LINE DESCRIPTOR BLOCK
209 ADD #4, (SP) ; OFFSET OF FLAG WORD
210 BITB #FL$SWT, @ (SP)+ ; IF THIS MODEM SWITCHES CARRIER
211 BEQ 10$; THEN
212 BISB #MS.SYN, M.STS(R5) ; MONITOR DATA SET READY ONLY
213 : IF
214 BICB #MS.SWT, M.STS(R5) ; Assume Carrier and DSR to be monitored
215 MOV @SPAR1, -(SP) ; Address of line descriptor block
216 ADD #4, (SP) ; Offset of flag word
217 BITB #FL$SWT, @ (SP)+ ; If this modem switches Carrier
218 BEQ 10$; then
219 BISB #MS.SWT, M.STS(R5) ; Monitor Data Set Ready only
220 .ENDC
221 .ENDC
222 10$: RETURN
223 ;

```

```

590 .SBTTL POST ASYNCHRONOUS DISCONNECT
591 :+
592 **--DISC-POST ASYNCHRONOUS DISCONNECT
593 :
594 CARRIER (ASYNCR) OR DATASET READY (SYNCR) WAS LOST WHILE THE LINE
595 WAS ACTIVE. POST AN ASYNCHRONOUS DISCONNECT COMPLETION TO THE
596 DLC PROCESS.
597 :-
598 INPUTS:
599 R4 - SYSTEM LINE NUMBER
600 R5 - POINTER TO LINE ENTRY IN MDC DATABASE
601 :
602 000756 112765 000014 000000 DISC: MOVB #ST.CER,M.STT(R5) ; SET NEW STATE ... CONNECT ERROR
603 000764 012703 101366 MOV #CS.FIL!CE.DIS,R3 ; SET COMPLETION STATUS
604 000770 CALL @DDAST ; POST ASYNCHRONOUS COMPLETION
605 000774 103003 BCC 10$; SUCCESSFUL?
606 000776 152765 000200 000004 BISB #MC.CCB,M.CSV(R5) ; NO ... MARK ALLOCATION FAILURE
607 001004 10$: RETURN
608 :

```

AXSCH MACRO V05.03b Friday 28-Jun-85 18:30  
Table of contents

|    |    |                                              |
|----|----|----------------------------------------------|
| 5- | 53 | Macro definitions                            |
| 6- | 71 | \$SQSRV - GENERALIZED SOFTWARE LEVEL SERVICE |

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49

```
.IIF NDF K$$$DAS .TITLE AXSUB
.IIF DF K$$$DAS .TITLE AXSUBD
.IDENT /V05.00/
.ENABL LC
```

```
:+
: Copyright (c) 1981, 1982, 1983, 1985 by
: Digital Equipment Corporation, Maynard, Mass.
```

```
: This software is furnished under a license and may be used and copied
: only in accordance with the terms of such license and with the
: inclusion of the above copyright notice. This software or any other
: copies thereof may not be provided or otherwise made available to any
: other person. No title to and ownership of the software is hereby
: transferred.
```

```
: The information in this software is subject to change without notice
: and should not be construed as a commitment by Digital Equipment
: Corporation.
```

```
: Digital assumes no responsibility for the use or reliability of its
: software on equipment which is not supplied by Digital.
```

```
: Module Description
```

```
: CEX subroutines (resident in AUX)
```

```
: Distributed Systems Software Engineering
```

```
: Ident History:
```

```
: 1.00 10-Nov-80
:
: 3.00 16-Apr-82
: DECnet-11M V3.1
: DECnet-11M-PLUS V1.1
:
: 4.00 07-NOV-83
: DECNET-11M V4.0
: DECNET-11M-PLUS V2.0
:
: 5.00 22-JUL-85
: DECnet-11M/S V4.2
: DECnet-11M-Plus V3.0
: DECnet-Micro/RSX V1.0
```



AXSUB MACRO V05.03b Friday 28-Jun-85 18:30 Page 12-2  
Symbol table

|                  |                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|------------------|
| ALOCB = ***** GX | CS.BUF= 000200   | C.XPLI 000040    | FS.SFR= 006000   | R\$\$DER= 000000 |
| A\$\$CHK= 000000 | CS.CES= 000002   | C.XPT 000034     | FS.SFS= 004000   | R\$\$K11= 000001 |
| A\$\$CPS= 000000 | CS.CHN= 000010   | C.XSVC 000042    | FS.SPW= 040000   | R\$\$SND= 000000 |
| A\$\$PRI= 000000 | CS.CMP= 000200   | C.XTC 000037     | FS.STM= 000000   | R\$\$11M= 000000 |
| A\$\$TRP= 000000 | CS.DCR= 000400   | C.X25 000036     | FS.STP= 002000   | SQRCM = ***** GX |
| CB.CCB= 000002   | CS.DEF= 000004   | DEACB = ***** GX | FS.STR= 001000   | SRSTD = ***** GX |
| CB.DDM= 000040   | CS.DEV= 000002   | D\$\$BUG= 177514 | FS.TRM= 003000   | S\$\$WRG= 000000 |
| CB.DLC= 000020   | CS.DIS= 000040   | D\$\$ISK= 000000 | FS.WLB= 001000   | S\$\$YSZ= 007600 |
| CB.RDB= 000004   | CS.ENA= 000001   | D\$\$L11= 000001 | FS.XKL= 002000   | T\$\$KMG= 000000 |
| CB.SDB= 000010   | CS.ENB= 000020   | D\$\$YNC= 000000 | FS.XOF= 010000   | T\$\$MIN= 000000 |
| CB.SLI= 000100   | CS.ERR= 100000   | D\$\$YNM= 000000 | FS.XON= 007000   | V\$\$CTR= 001000 |
| CB.XLB= 000001   | CS.FTL= 001000   | EXRQF = ***** GX | FS.ZER= 002000   | XAVL = ***** GX  |
| CCBGT = ***** GX | CS.HCR= 000001   | E\$\$XPR= 000000 | F\$\$LVL= 000001 | XAVLL 00002R     |
| CC.LLC= 000200   | CS.HFE= 002000   | F.C.CP= 000020   | F2.DAS= ***** GX | X\$\$DBT= 000000 |
| CEACC = ***** GX | CS.LST= 040000   | FC.CTL= 000006   | G\$\$1PP= 000000 | ZF.COJ= 001000   |
| CEDIV = ***** GX | CS.MTL= 004000   | FC.KCP= 000016   | G\$\$TSS= 000000 | ZF.DDM= 000001   |
| CE.ABO= 100362   | CS.RNG= 000010   | FC.KIL= 000004   | G\$\$TTK= 000000 | ZF.DIA= 004000   |
| CE.DAO= 100346   | CS.ROV= 000004   | FC.MAN= 000024   | G\$\$WRD= 000000 | ZI.DLC= 000002   |
| CE.DIS= 100366   | CS.RSN= 010000   | FC.MLD= 000026   | I\$\$RAR= 000000 | ZF.DVP= 100000   |
| CE.ERR= 100370   | CS.SHU= 000001   | FC.PCT= 000030   | I\$\$RDN= 000000 | ZF.INI= 040000   |
| CE.ILN= 100350   | CS.SID= 000002   | FC.PWR= 000022   | KISAR6= ***** GX | ZF.KMX= 000020   |
| CE.LTO= 100356   | CS.STR= 000004   | FC.RCE= 000002   | K\$\$CNT= 177546 | ZF.LLC= 000004   |
| CE.MOP= 100372   | CS.SUC= 000001   | FC.RCP= 000014   | K\$\$CSR= 177546 | ZF.LMC= 000100   |
| CE.NTE= 100361   | CS.TMO= 020000   | FC.TIM= 000010   | K\$\$LDC= 000000 | ZF.MAN= 020000   |
| CE.RTE= 100376   | CS.XUR= 000004   | FC.XCP= 000012   | K\$\$TPS= 000074 | ZF.MFL= 000010   |
| CE.SRC= 100364   | C\$\$CKP= 000000 | FC.XME= 000000   | LDL P = 000000   | ZF.MTM= 000400   |
| CE.STP= 100372   | C\$\$ORE= 000400 | FMSK2 = ***** GX | L\$\$ASG= 000000 | ZF.MUX= 000040   |
| CE.TME= 100354   | C\$\$RSH= 177564 | FS.AST= 000000   | L\$\$DRV= 000000 | ZF.PSE= 002000   |
| CE.TMO= 100374   | C.ADD 000034     | FS.CIB= 002000   | L\$\$P11= 000001 | ZF.SLI= 010000   |
| CE.UNS= 100344   | C.BID 000003     | FS.CRA= 001000   | L\$\$11R= 000000 | ZF.TIM= 000200   |
| C..CHN= 000001   | C.BUF 000014     | FS.DIS= 013000   | MAXLEN= 000006   | ZF.X3P= 000000   |
| CF.EOM= 000004   | C.BUF1 000014    | FS.DVC= 001000   | MLDTSK 000014R   | ZS.ASN= 100000   |
| CF.HDR= 000C20   | C.BUF2 000024    | FS.ENB= 012000   | M\$\$CRB= 000124 | ZS.BSY= 140000   |
| CF.LB= 100000    | C.CNT 000020     | FS.EXI= 001000   | M\$\$CRX= 000000 | Z.AVL 000314     |
| CF.LIN= 000002   | C.CNT1 000020    | FS.GET= 006000   | M\$\$FCS= 000000 | Z.DAT 000016     |
| CF.SOM= 000010   | C.CNT2 000030    | FS.HLT= 000000   | M\$\$MGE= 000000 | Z.DSP 000000     |
| CF.SYN= 000040   | C.FLG 000022     | FS.INI= 000000   | M\$\$NET= 000000 | Z.FLG 000010     |
| CF.TRN= 000100   | C.FLG1 000022    | FS.KIL= 000000   | M\$\$QVR= 000000 | Z.LEN = 000016   |
| CM.CIR= 000002   | C.FLG2 000032    | FS.LCL= 100000   | N\$\$ACC= 003001 | Z.LLN = 000006   |
| CM.FMT= 100000   | C.FNC 000010     | FS.LTM= 001000   | N\$\$BUF= 000001 | Z.MAP 000020     |
| CM.HRD= 00C002   | C.LIN 000006     | FS.MNT= 004000   | N\$\$LDV= 000001 | Z.NAM 000004     |
| CM.LIN= 000000   | C.LNK 000000     | FS.MSN= 014000   | N\$\$MCP= 000001 | Z.PCB 000012     |
| CM.LOO= 000001   | C.MOD 000011     | FS.REA= 001000   | N\$\$MLL= 000001 | Z.SCH 000007     |
| CM.XLO= 000004   | C.NSP 000004     | FS.RET= 000000   | N\$\$MOV= 000010 | \$ALOCX 000020RG |
| CP.DCF= 000040   | C.PRO 000042     | FS.REZ= 003000   | N\$\$NCT= 000001 | \$CMEXI 000602RG |
| CP.HDL= 000007   | C.RSV 000002     | FS.RLB= 020000   | N\$\$PEM= 000001 | \$CMINI 000570RG |
| CP.PSI= 177400   | C.STA 000007     | FS.RNG= 011000   | P\$\$P45= 000000 | \$DEACX 000150RG |
| CP.PSI= 000200   | C.STS 000012     | FS.RST= 000000   | P\$\$WRD= 000000 | \$ROSTL 000650RG |
| CP.XCF= 000100   | C.URM 177776     | FS.RTN= 001000   | Q\$\$OPT= 000010 | \$SQRTB 000006R  |
| CP.2FR= 000030   | C.XACP 000004    | FS.SET= 005000   | RDBNM = ***** GX | \$XLINK 000432RG |
| CS.ABO= 000100   | C.XID 000035     | FS.SFC= 005000   | RDBTH = ***** GX | .\$\$\$.= 000034 |
| CS.BRO= 000002   | C.XLEN 000044    |                  |                  |                  |

. ABS. 177776 000 (RW,I,BGL,ABS,OVR)  
001006 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

AXTIM - AUXILLIARY PROCESS TIME MACRO V05.03b Friday 28-Jun-85 18:31 <sup>N. 6</sup> Page 8-2  
PERFORM ONCE-PER-SECOND OPERATIONS

```
266 BIT #F2.MP, @FMSK2 : IS THIS A MULTI-PROCESSOR?
267 BEQ 80$: BR IF NO
268 TST (SP)+ : CLEAN UP STACK
269
270 .ENDC
271
272 .IFTF
273
274 000240 005726 80$: TST (SP)+ : CLEAN UP STACK
275
276 .ENDC
277
278 000242 RETURN
```

|          |            |          |          |        |    |      |      |
|----------|------------|----------|----------|--------|----|------|------|
| SSSSSSSS | EEEEEEEEEE | SSSSSSSS | CCCCCCCC | 000000 | NN | NN   |      |
| SSSSSSSS | EEEEEEEEEE | SSSSSSSS | CCCCCCCC | 000000 | NN | NN   |      |
| SS       | EE         | SS       | CC       | 00     | 00 | NN   | NN   |
| SS       | EE         | SS       | CC       | 00     | 00 | NN   | NN   |
| SS       | EE         | SS       | CC       | 00     | 00 | NNNN | NN   |
| SS       | EE         | SS       | CC       | 00     | 00 | NNNN | NN   |
| SSSSSS   | EEEEEEEE   | SSSSSS   | CC       | 00     | 00 | NN   | NN   |
| SSSSSS   | EEEEEEEE   | SSSSSS   | CC       | 00     | 00 | NN   | NN   |
| SS       | EE         | SS       | CC       | 00     | 00 | NN   | NNNN |
| SS       | EE         | SS       | CC       | 00     | 00 | NN   | NNNN |
| SS       | EE         | SS       | CC       | 00     | 00 | NN   | NN   |
| SS       | EE         | SS       | CC       | 00     | 00 | NN   | NN   |
| SSSSSSSS | EEEEEEEEEE | SSSSSSSS | CCCCCCCC | 000000 | NN | NN   | .... |
| SSSSSSSS | EEEEEEEEEE | SSSSSSSS | CCCCCCCC | 000000 | NN | NN   | .... |

|        |        |          |
|--------|--------|----------|
| 11     | 11     | SSSSSSSS |
| 11     | 11     | SSSSSSSS |
| 1111   | 1111   | SS       |
| 1111   | 1111   | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SSSSSS   |
| 11     | 11     | SSSSSS   |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 111111 | 111111 | SSSSSSSS |
| 111111 | 111111 | SSSSSSSS |

|                  |                 |                 |                 |                  |
|------------------|-----------------|-----------------|-----------------|------------------|
| LT.DIR= 000010   | MA.CI = 000040  | NE\$TPA= 000010 | N\$MBXQ 000050  | N.SND 000030     |
| LT.LCL= 000001   | MA.DA = 000000  | NE\$UDB= 000004 | N\$PLLT 000030  | N.SNM 000046     |
| LT.LPL= 000002   | MA.IL = 00002C  | NF\$ACC= 100000 | N\$SLA 000016   | N.SNMC 000044    |
| LT.NOT= 000040   | MC.CC = 000040  | NF\$BLK= 000100 | N\$SNOD 000012  | N.SOT 000037     |
| LT.RSU= 000200   | MC.CI = 000020  | NF\$DMO= 000010 | N\$TIM 000004   | N.SUS 000042     |
| LT.SLI= 000004   | MC.DC = 000100  | NF\$EVT= 040000 | N\$VCB 000010   | OF.PRO= 000040   |
| LT.TDA= 000100   | MC.DI = 000060  | NF\$MOU= 000040 | N\$SACC= 000001 | OF.RLU= 000100   |
| L\$ASG= 000000   | MC.NO = 000000  | NF\$RST= 000002 | N\$SEVL= 000001 | OF.SMC= 000200   |
| L\$DRV= 000000   | MC.RC = 000140  | NF\$SCN= 000020 | N\$SLDV= 000001 | O.FLG 000003     |
| L\$P11= 000001   | MD.BM = 000040  | NF\$SHU= 000004 | N\$SMLL= 000001 | O.LEN 000012     |
| L\$11R= 000000   | MD.EM = 000100  | NF\$TIM= 000200 | N\$SMOV= 000010 | O.LNK 000000     |
| L.CSTA 000037    | MD.ILS= 000040  | NF.ACC= 000001  | N\$SNCT= 000001 | O.MXC 000004     |
| L.CTR 000074     | MD.IM = 000020  | NF.BCP= 000040  | N\$SPEM= 000001 | O.NAM 000006     |
| L.DCR 000100     | MF.ACK= 000004  | NF.END= 000200  | N\$SES= 000001  | O.TYP 000002     |
| L.FLAG 000014    | MF.CTL= 000010  | NF.EVT= 000002  | N.CAC 000120    | P\$SP45= 000000  |
| L.ILSQ 000052    | MF.DAT= 000000  | NF.LLI= 000004  | N.CACC 000116   | P\$SWRD= 000000  |
| L.ILTT 000066    | M\$CRB= 000124  | NF.LV2= 000100  | N.CBL = 000142  | QUETSK 000716RG  |
| L.LDA 000032     | M\$CRX= 000000  | NF.SLI= 000400  | N.CDA 000142    | Q\$SOP= 000010   |
| L.LIA 000034     | M\$FCS= 000000  | NF.SMC= 000020  | N.CDAC 000140   | RMVWND= ***** GX |
| L.LLA 000002     | M\$MGE= 000000  | NF.SWP= 000010  | N.CDDS 000070   | R\$SDER= 000000  |
| L.LNG 000124     | M\$MUP= 000000  | NO.DTR= 000077  | N.CDEV 000062   | R\$SK11= 000001  |
| L.LNO 000026     | M\$NET= 000000  | NO.FAL= 000021  | N.CID 000064    | R\$SSND= 000000  |
| L.LPT 000065     | M\$SOVR= 000000 | NO.FAI= 000001  | N.CIDC 000062   | R\$11M= 000000   |
| L.LSA 000030     | M.MAIL 000014   | NO.NCU= 000023  | N.CPS 000106    | R\$11S= 000000   |
| L.LSPD 000046    | M.MAX 000011    | NO.RTL= 000022  | N.CPSC 000104   | SNDACC= ***** GX |
| L.LSFI 000044    | M.MBL 000020    | NO.TAS= 000000  | N.CTL 000000    | SNDCON= ***** GX |
| L.LTT 000062     | M.NAST 000007   | NO.TCL= 000017  | N.CUIC 000066   | SRSTD = ***** GX |
| L.MASQ 000070    | M.NEXT 000002   | NO.TC1= 000005  | N.CUNI 000064   | ST\$CC = 000004  |
| L.MAST 000073    | M.RESP 000016   | NO.TLK= 000020  | N.DDE 000010    | ST\$CIR= 000006  |
| L.MASZ 000072    | M.SPA 000012    | NS\$DON= 000000 | N.DDEC 000006   | ST\$CIS= 000002  |
| L.NIN 000020     | M.TASK 000004   | NS\$SDI= 000002 | N.DFM 000004    | ST\$DAT= 000010  |
| L.NXN 000016     | M.USE 000010    | NS\$WDC= 000004 | N.DGP 000006    | ST\$DIP= 000012  |
| L.NXTH 000010    | NC.FMO= 000000  | NT.ABO= 000005  | N.DNM 000014    | ST\$PND= 000014  |
| L.OPD 000103     | NC.FM1= 000001  | NT.ABT= 000004  | N.DNMC 000012   | S\$SWRG= 000000  |
| L.OPDL 000102    | NC.FM2= 000002  | NT.ABT= 000001  | N.DOT 000005    | S\$YSZ= 007600   |
| L.REM 000006     | NE\$ABM= 000010 | NT.DSC= 000003  | N.DUS 000010    | TLACHK= ***** GX |
| L.RFC 000050     | NE\$ABO= 000046 | NT.EVT= 000006  | N.RAC 000070    | T\$KMG= 000000   |
| L.RLA 000004     | NE\$ACC= 000042 | NT.'NT= 000002  | N.RACC 000066   | T\$MIN= 000000   |
| L.RNO 000022     | NE\$ACT= 000042 | NT.MOP= 000010  | N.RDE 000012    | T.NAM = ***** GX |
| L.RTO 000060     | NE\$COM= 000047 | NT.NSP= 000010  | N.RDEC 000010   | UISAR6= ***** GX |
| L.RTYD 000055    | NE\$FCF= 000050 | NT.VFY= 000007  | N.RFM 000006    | USRCI 000360RG   |
| L.RTYI 000057    | NE\$FMT= 000005 | NS\$ACO 000000  | N.RGP 000010    | US\$CNF= 000002  |
| L.SEC 000064     | NE\$GEN= 000007 | NS\$ACTL 000032 | N.RID 000034    | US\$DIS= 000006  |
| L.SEGZ 000076    | NE\$IFC= 000030 | NS\$CIR 000033  | N.RIDC 000032   | US\$DON= 000000  |
| L.STA 000000     | NE\$ILS= 000043 | NS\$DLA 000020  | N.RND 000000    | US\$DSC= 000004  |
| L.TC 000042      | NE\$IMG= 000042 | NS\$DLY 000014  | N.RNM 000016    | US\$EAC= 000012  |
| L.TIC 000043     | NE\$MLB= 000006 | N.ELEN 000054   | N.RNMC 000014   | US\$WDS= 000010  |
| L.TIPD 000013    | NE\$CNP= 000012 | NS\$ENC 000042  | N.ROT 000007    | U.CW3 = ***** GX |
| L.TIPI 000012    | NE\$NOD= 000002 | NS\$ERRC 000022 | N.RPS 000056    | VE.FAI= 177777   |
| L.TMRD 000054    | NE\$NSD= 000003 | NS\$FLG 000005  | N.RPSC 000054   | VFNAM= ***** GX  |
| L.TMRI 000056    | NE\$NSI= 000013 | NS\$FNC 000006  | N.RQL = 000110  | VS.NPV= 000001   |
| L.TYP 000001     | NE\$NSR= 000003 | NS\$GENQ 000052 | N.RUS 000012    | VS.PRV= 000002   |
| L.USA 000024     | NE\$RES= 000001 | NS\$GTM 000015  | N.SDE 000042    | VZ.NVD= 000000   |
| L.USTA 000036    | NE\$SSR= 000000 | NS\$HIGH 000033 | N.SDEC 000040   | V\$CTR= 001000   |
| L.VER 000015     | NE\$SSS= 000045 | N\$LLT 000026   | N.SEGZ 000002   | WK.ACK= 000001   |
| L.WIND 000040    | NE\$TCN= 000040 | N\$LLTM 000024  | N.SFM 000036    | WK.AST= 000200   |
| MAPOBJ= ***** GX | NE\$TCO= 000006 | N\$LVLC 000036  | N.SGP 000040    | WK.DIS= 000010   |

```

226 000100 016467 000010 000000G MOV W.TMP(R4), $IOPKT; Is there a pending disconnect request?
227 000106 001404 BEQ 20$; If EQ, no
228 000110 012700 000000C MOV #IE.AB0&377,R0 ; Complete the request in error
229 000114 CALL IODUN1 ; ...
230
231 000120 012603 20$: MOV (SP)+,R3 ; Recover LLi address
232 000122 CALL ACCLLT ; Gain access to the LLT
233
234 000132 012763 000046 000100 MOV #ER$AB0.1.DCR(R3)
235 000140 012702 000264 MOV #CLSTA-2,R2 ; Point to disconnect substate table
236 000144 CALL KILLNK ; Kill the logical link
237
238 000150 RESRG <R4,R3,R1> ; Recover registers
239
240 000156 CALL RMVWND ; Release the window block resources
241 000162 105364 000010 DECB M.USE(R4) ; One less link in use
242
243 000166 005721 100$: TST (R1)+ ; Skip LUT entry
244 000170 005316 DEC (SP) ; Any more to go?
245 000172 001321 BNE 10$; If NE, yes
246 000174 005726 TST (SP)+ ; Clean up stack
247
248
249 ;+
250 ;**--CLSDON-Completion of network deaccess
251 ;
252 ; This routine is called to complete processing of a network deaccess
253 ; request.
254 ;
255 ; Inputs:
256 ; R4 = Address of mailbox
257 ; R5 = Address of database descriptor
258
259 000176 010' 3 CLSDON: MOV R5,R0 ; Compute address of mailbox listhead
260 000200 062' 0 ADD #N$MBXQ-M.NEXT,R0
261
262 000204 026004 00000? 10$: CMP M.NEXT(R0),R4 ; Scan mailbox queue for this entry
263 000210 001403 BEQ 20$; If EQ, found
264 000212 016000 000002 MOV M.NEXT(R0),R0 ; Move down list
265 000216 000772 BR 10$; and try again
266
267 000220 016460 000002 000002 20$: MOV M.NEXT(R4),M.NEXT(R0)
268 000226 016403 000012 MOV M.SPA(R4),R3 ; Recover I/O packet address
269 000232 010367 000000G MOV R3,$IOPKT ; Save it for later I/O completion
270 000236 RESRG <R1> ; Recover task header address
271
272 .IF DF X$$HDR
273 MAP $HDRMP ; Map to the task's header
274
275 .ENDC
276
277 000240 005073 000000G CLR @I.LN2(R3) ; Remove address of mailbox from LUT table
278 000244 105061 000000G CLRB H.NML(R1) ; Network no longer accessed
279
280 000250 010400 MOV R4,R0 ; Copy address of mailbox
281 000252 012701 000020 MOV #M.MBL,R1 ; Size of mailbox
282 000256 CALL @DEACB ; Deallocate the mailbox

```

```

700 .SBTTL Access control completion
701
702 + **NVP-Access control completion
703
704 This routine handles the access control completion from the network
705 verification program (NVP...).
706
707 Inputs:
708 R1 = Address of the task's header
709 R2 = Subfunction code/4
710 R3 = Address of I/O packet
711 R4 = Address of mailbox
712 R5 = Address of database descriptor
713 Task header is mapped (RSX-11M-Plus ONLY)
714
715 .IF DF N$SACC
716
717 NVP: MAP I.PRM(R3) ; Map to the user buffer
718 MOV I.PRM+2(R3),R0 ; Get virtual address of user's buffer
719 MOV N.CTL+4(R0),I.PRM+16(R3)
720
721 MOV R4,$MAIBX ; Save address of mailbox
722 CALL TLCHK ; Check for valid TLA
723 BCS BADERR ; If CS, no match
724
725 DECB M.USE(R4) ; Reduce count of active links
726 MOV $IOPKT,R3 ; Recover I/O packet address
727 MOV I.PRM+6(R3),C.FLG2(R1)
728
729 MOV R1,-(SP) ; Save verification CCB
730 MOV R1,R0 ; Copy CCB address
731 MOV I.PRM(R3),R1 ; Set source APR bias
732 MOV I.PRM+2(R3),R2 ; and source virtual address
733 ADD #N.CIDC+4-2000,R2
734 MOV C.BUF(R0),R3 ; Set destination APR bias
735 MOV C.BUF+2(R0),R4 ; and destination virtual address
736 ADD #N.CIDC,R4 ; Start at source task ID byte count
737 MOV #N.CACC-N.CIDC,R0
738 CALL @BLXIO ; Copy the verification results
739
740 CALL IOSUC ; Complete the I/O request successfully
741
742 MOV (SP)+,R4 ; Recover verification CCB
743 MOV #NT.CON!CM.CON,C.FNC(R4)
744 BICB #CX.UNL,C.MOD(R4)
745 MOV #ERACC,NERRC(R5) ; Set up 'access control failure' reason
746
747 .IF NDF RSPRO ; On PRO/DENet, validate all verif levels.
748 MOV C.FLG2+1(R4),R0 ; Get requested verification level
749 BNE 10$; If NE, allow connect to pass
750
751 .IFF ; NDF RSPRO
752 TST C.PRO(R4) ; Did a virtual terminal get setup??
753 BEQ 20$; BR if not, reject the connection.
754 TSTB C.FLG2(R4) ; Did verification succeed?
755 BLT 20$; If LE, no ... reject the connection
756 ; (note that for PRO/DENet, no verification

```

```

42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,SAVMAP,RESMAP,MAPLLT
45 .MCALL CCBDF$,CTRDF$,ECDDB$,LLTDF$,EVLDF$
46
47 000000 CCBDF$; Define CCB offsets
48 000000 CTRDF$; Define counter block offsets
49 000000 ECDDB$; Define ECL database offsets
50 000000 LLTDF$; Define LLT offsets
51 000000 EVLDF$; Define event logger symbols
52
53 000C01 N$$SES = 1 ; This module is part of session control

```

FILE ID SEDAT

N 12

|          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|
| SSSSSSSS | EEEEEEEE | SSSSSSSS | DDDDDDDD | AAAAAA   | TTTTTTTT |
| SSSSSSSS | EEEEEEEE | SSSSSSSS | DDDDDDDD | AAAAAA   | TTTTTTTT |
| SS       | EE       | SS       | DD DD    | AA AA    | TT       |
| SS       | EE       | SS       | DD DD    | AA AA    | TT       |
| SS       | EE       | SS       | DD DD    | AA AA    | TT       |
| SS       | EE       | SS       | DD DD    | AA AA    | TT       |
| SSSSSS   | EEEEEEEE | SSSSSS   | DD DD    | AA AA    | TT       |
| SSSSSS   | EEEEEEEE | SSSSSS   | DD DD    | AA AA    | TT       |
| SS       | EE       | SS       | DD DD    | AAAAAAAA | TT       |
| SS       | EE       | SS       | DD DD    | AAAAAAAA | TT       |
| SS       | EE       | SS       | DD DD    | AA AA    | TT       |
| SS       | EE       | SS       | DD DD    | AA AA    | TT       |
| SSSSSS   | EEEEEEEE | SSSSSS   | DDDDDD   | AA AA    | TT       |
| SSSSSS   | EEEEEEEE | SSSSSS   | DDDDDD   | AA AA    | TT       |

....  
....  
....  
....

|        |        |          |
|--------|--------|----------|
| 11     | 11     | SSSSSSSS |
| 11     | 11     | SSSSSSSS |
| 1111   | 1111   | SS       |
| 1111   | 1111   | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SSSSSS   |
| 11     | 11     | SSSSSS   |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 111111 | 111111 | SSSSSSSS |
| 111111 | 111111 | SSSSSSSS |



SES DAT11S CREATED BY MACRO ON 28-JUN-85 AT 19:55 PAGE 4 N 13

MACRO CROSS REFERENCE

CREF 04.00

MACRO NAME REFERENCES

CRBDF\$ #6-44 6-46

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

```
.TITLE SESDMO - Session control dismount processing
.IDENT /V05.00/
.ENABL LC

: Copyright (C) 1982, 1983, 1985 by
: Digital Equipment Corporation, Maynard, MASS.

: This software is furnished under a license for use only on a
: single computer system and may be copied only with the
: inclusion of the above copyright notice. This software, or
: any other copies thereof, may not be provided or otherwise
: made available to any other person except for use on such
: system and to one who agrees to these license terms. Title
: to and ownership of the software shall at all times remain
: in DEC.

: The information in this document is subject to change without
: notice and should not be construed as a commitment by Digital
: Equipment Corporation.

: DEC assumes no responsibility for the use or reliability of
: its software on equipment which is not supplied by DEC.

: Module description

: Session control dismount processing

: Ident history:

: 4.00 07-NOV-83
: DECNET-11M V4.0
: DECNET-11M-PLUS V2.0

: 5.00 22-JUL-85
: DECnet-11M/s V4.2
: DECnet-11M-Plus V3.0
: DECnet-Micro/RX V1.0
:
```

SESDSP - Session control dispat MACRO V05.03b Friday 28-Jun-85 19:56 <sup>N 15</sup>  
Table of contents

|     |     |                                        |
|-----|-----|----------------------------------------|
| 6-  | 42  | Macro definitions                      |
| 7-  | 56  | Dispatch tables                        |
| 8-  | 98  | Session control ACP idle loop          |
| 9-  | 223 | Scan general delivery queue            |
| 10- | 272 | Request task to run (from AUX request) |
| 11- | 339 | Request task to run                    |
| 12- | 466 | Determine task name                    |
| 15- | 639 | Complete I/O request                   |

B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z

[illegible]

|   | 5  | 6  | 7  | 8  | 9  | 10 |
|---|----|----|----|----|----|----|
| J | 5  | 6  | 6  | 6  | 6  | 6  |
| K | 5  | 6  | 6  | 6  | 6  | 6  |
| L | 5  | 6  | 6  | 6  | 6  | 6  |
| M | 5  | 6  | 6  | 6  | 6  | 6  |
| N | 5  | 6  | 6  | 6  | 6  | 6  |
| B | 6  | 6  | 6  | 6  | 6  | 6  |
| C | 6  | 6  | 6  | 6  | 6  | 6  |
| D | 6  | 6  | 6  | 6  | 6  | 6  |
| E | 6  | 6  | 6  | 6  | 6  | 6  |
| F | 6  | 6  | 6  | 6  | 6  | 6  |
| G | 6  | 6  | 6  | 6  | 6  | 6  |
| H | 6  | 6  | 6  | 6  | 6  | 6  |
| I | 6  | 6  | 6  | 6  | 6  | 6  |
| J | 6  | 6  | 6  | 6  | 6  | 6  |
| K | 6  | 6  | 6  | 6  | 6  | 6  |
| L | 6  | 6  | 6  | 6  | 6  | 6  |
| M | 6  | 6  | 6  | 6  | 6  | 6  |
| N | 6  | 6  | 6  | 6  | 6  | 6  |
| B | 7  | 7  | 7  | 7  | 7  | 7  |
| C | 7  | 7  | 7  | 7  | 7  | 7  |
| D | 7  | 7  | 7  | 7  | 7  | 7  |
| E | 7  | 7  | 7  | 7  | 7  | 7  |
| F | 7  | 7  | 7  | 7  | 7  | 7  |
| G | 7  | 7  | 7  | 7  | 7  | 7  |
| H | 7  | 7  | 7  | 7  | 7  | 7  |
| I | 7  | 7  | 7  | 7  | 7  | 7  |
| J | 7  | 7  | 7  | 7  | 7  | 7  |
| K | 7  | 7  | 7  | 7  | 7  | 7  |
| L | 7  | 7  | 7  | 7  | 7  | 7  |
| M | 7  | 7  | 7  | 7  | 7  | 7  |
| N | 7  | 7  | 7  | 7  | 7  | 7  |
| B | 8  | 8  | 8  | 8  | 8  | 8  |
| C | 8  | 8  | 8  | 8  | 8  | 8  |
| D | 8  | 8  | 8  | 8  | 8  | 8  |
| E | 8  | 8  | 8  | 8  | 8  | 8  |
| F | 8  | 8  | 8  | 8  | 8  | 8  |
| G | 8  | 8  | 8  | 8  | 8  | 8  |
| H | 8  | 8  | 8  | 8  | 8  | 8  |
| I | 8  | 8  | 8  | 8  | 8  | 8  |
| J | 8  | 8  | 8  | 8  | 8  | 8  |
| K | 8  | 8  | 8  | 8  | 8  | 8  |
| L | 8  | 8  | 8  | 8  | 8  | 8  |
| M | 8  | 8  | 8  | 8  | 8  | 8  |
| N | 8  | 8  | 8  | 8  | 8  | 8  |
| B | 9  | 9  | 9  | 9  | 9  | 9  |
| C | 9  | 9  | 9  | 9  | 9  | 9  |
| D | 9  | 9  | 9  | 9  | 9  | 9  |
| E | 9  | 9  | 9  | 9  | 9  | 9  |
| F | 9  | 9  | 9  | 9  | 9  | 9  |
| G | 9  | 9  | 9  | 9  | 9  | 9  |
| H | 9  | 9  | 9  | 9  | 9  | 9  |
| I | 9  | 9  | 9  | 9  | 9  | 9  |
| J | 9  | 9  | 9  | 9  | 9  | 9  |
| K | 9  | 9  | 9  | 9  | 9  | 9  |
| L | 9  | 9  | 9  | 9  | 9  | 9  |
| M | 9  | 9  | 9  | 9  | 9  | 9  |
| N | 9  | 9  | 9  | 9  | 9  | 9  |
| B | 10 | 10 | 10 | 10 | 10 | 10 |
| C | 10 | 10 | 10 | 10 | 10 | 10 |
| D | 10 | 10 | 10 | 10 | 10 | 10 |

SESCON

|   |    |        |
|---|----|--------|
| E | 10 |        |
| F | 10 |        |
| G | 10 |        |
| H | 10 |        |
| I | 10 |        |
| J | 10 |        |
| K | 10 |        |
| L | 10 |        |
| M | 10 |        |
| N | 10 |        |
| B | 11 |        |
| C | 11 |        |
| D | 11 |        |
| E | 11 |        |
| F | 11 |        |
| G | 11 |        |
| H | 11 |        |
| I | 11 |        |
| J | 11 | SESCTR |
| K | 11 |        |
| L | 11 |        |
| M | 11 |        |
| N | 11 |        |
| B | 12 |        |
| C | 12 |        |
| D | 12 |        |
| E | 12 |        |
| F | 12 |        |
| G | 12 |        |
| H | 12 |        |
| I | 12 |        |
| J | 12 |        |
| K | 12 |        |
| L | 12 |        |
| M | 12 | SESDAT |
| N | 12 |        |
| B | 13 |        |
| C | 13 |        |
| D | 13 |        |
| E | 13 |        |
| F | 13 |        |
| G | 13 |        |
| H | 13 |        |
| I | 13 |        |
| J | 13 |        |
| K | 13 |        |
| L | 13 |        |
| M | 13 |        |
| N | 13 |        |
| B | 14 | SESDis |
| C | 14 |        |
| D | 14 |        |
| E | 14 |        |
| F | 14 |        |
| G | 14 |        |
| H | 14 |        |
| I | 14 |        |
| J | 14 |        |
| K | 14 | SESDMO |
| L | 14 |        |

SESDAT

SESDMO

|   | SES | SDSP |
|---|-----|------|
| M | 14  |      |
| N | 14  |      |
| B | 15  |      |
| C | 15  |      |
| D | 15  |      |
| E | 15  |      |
| F | 15  |      |
| G | 15  |      |
| H | 15  |      |
| I | 15  |      |
| J | 15  |      |
| K | 15  |      |
| L | 15  |      |
| M | 15  |      |
| N | 15  |      |
| B | 16  |      |
| C | 16  |      |
| D | 16  |      |
| E | 16  |      |
| F | 16  |      |
| G | 16  |      |
| H | 16  |      |
| I | 16  |      |
| J | 16  |      |
| K | 16  |      |
| L | 16  |      |
| M | 16  |      |

```

466 .SBTTL Determine task name
467 ;+
468 ;**--NMXXX-Determine task name
469 ;
470 Determine the task name to be used for the task request logic.
471 ;
472 ;Inouts:
473 R4 = Address of CCB
474 R5 = Address of database descriptor
475 ;
476 ;Outputs:
477 R3 = Pointer to task name
478 ;
479 ;Registers modified:
480 R1
481 ;
482 ;+
483 ;**--NMCON-Connect request
484 ;
485 NMCON: .IF DF N$$MCP
486 000370 MOV C.STS(R4), $RQCPY; Set max # of copies allowed
487 000374 .ENDC
488 016403 MOV C.BUF2+2(R4), R3 ; Get pointer to task name
489 000026 RETURN
490
491 ;+
492 ;**--NMEVT-Network event
493 ;
494 NMEVT: .IF DF N$$MCP
495 000376 MOV #N$$SMC, $RQCPY ; Set max # of copies allowed
496 010403 .ENDC
497 062703 MOV R4, R3 ; Compute address of task name
498 000400 ADD #C.CNT2, R3 ; ...
499 000404 RETURN
500
501 ;+
502 ;**--NMVFY-Verification request
503 ;
504 .IF DF N$$ACC
505 000406 NMVFY: MOV #VFYNAM, R3 ; Point to verification task name
506 012703 RETURN
507 000000G .ENDC
508
509
510
511
512
513
514
515
516
517
518

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

```
.TITLE SESINI - Session control initialisation
.IDENT /V05.00/
.ENABL LC
```

```
Copyright (C) 1982, 1983, 1985 by
Digital Equipment Corporation, Maynard, MASS.
```

```
This software is furnished under a license for use only on a
single computer system and may be copied only with the
inclusion of the above copyright notice. This software, or
any other copies thereof, may not be provided or otherwise
made available to any other person except for use on such
system and to one who agrees to these license terms. Title
to and ownership of the software shall at all times remain
in DEC.
```

```
The information in this document is subject to change without
notice and should not be construed as a commitment by Digital
Equipment Corporation.
```

```
DEC assumes no responsibility for the use or reliability of
its software on equipment which is not supplied by DEC.
```

#### Module description

Session control initialisation

#### Ident history:

```
4.00 07-NOV-83
 DECNET-11M V4.0
 DECNET-11M-PLUS V2.0

5.00 22-JUL-85
 DECnet-11M/S V4.2
 DECnet-11M-Plus V3.0
 DECnet-Micro/R SX V1.0
```

SESSUB11S CREATED BY MACRO ON 28-JUN-85 AT 20:00 PAGE 6 B 15  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE    | REFERENCES                                                       |
|---------|----------|------------------------------------------------------------------|
| MOS25A  | = 000006 | #6-54                                                            |
| MOS25P  | = 000002 | #6-54                                                            |
| MOS25S  | = 000004 | #6-54                                                            |
| MOS29S  | = 000010 | #6-54                                                            |
| MVFBF   | = *****  | 10-274                                                           |
| MSHIGH  | = 000003 | #6-54                                                            |
| MS3100  | = 000000 | #6-54                                                            |
| MS3101  | = 000001 | #6-54                                                            |
| MS3102  | = 000002 | #6-54                                                            |
| MS3103  | = 000003 | #6-54 6-54                                                       |
| M.MAIL  | 000014   | 9-229 23-931 40-1585                                             |
| M.NEXT  | 000002   | 26-1065 26-1067                                                  |
| M.SPA   | 000012   | 9-236                                                            |
| M.TASK  | 000004   | 26-1069 38-1507                                                  |
| M.USE   | 000010   | *24-994                                                          |
| NF\$BLK | = 000100 | 28-1156                                                          |
| NF\$SCN | = 000020 | 7-95                                                             |
| NF\$TJM | = 000200 | 7-95                                                             |
| NM\$ARA | = 176000 | 17-612 17-615 17-618                                             |
| NOOP    | 003506   | R #41-1647 41-1666                                               |
| NSTATE  | 003510   | RG 41-1637 #41-1654                                              |
| NS\$DON | = 000000 | 12-374 41-1642                                                   |
| NS\$SDI | = 000002 | 12-375 12-376 12-377 12-379                                      |
| NT.VFY  | = 000007 | 40-1608                                                          |
| N\$ACT. | 000032   | *8-185 8-186 8-188 *35-1433                                      |
| N\$DLA  | 000020   | *8-176 27-1127 27-1128                                           |
| N\$DLY  | 000014   | *11-343                                                          |
| N\$ENC  | 000042   | 11-298 11-300 20-806 20-808 20-825 20-830 *20-831                |
| N\$ERRC | 000022   | *8-116 *17-620 *17-631 *27-1104                                  |
| N\$FLG  | 000005   | *7-95 *28-1156                                                   |
| N\$GENQ | 000052   | 7-83 7-90                                                        |
| N\$GTM  | 000015   | *7-93                                                            |
| N\$HIGH | 000033   | 8-186 *8-188                                                     |
| N\$LLT  | 000026   | *8-138 35-1388                                                   |
| N\$LLTM | 000024   | *8-139 10-259 10-270 11-347 19-786 31-1257 35-1387 42-1711       |
| N\$LVC  | 000036   | 8-117 8-118 8-171 35-1415                                        |
| N\$MBXQ | 000050   | 26-1065                                                          |
| N\$PLLT | 000030   | *8-134 8-170 8-193 27-1122 35-1420                               |
| N\$SLA  | 000016   | 27-1125 27-1126 11-337 15-467 17-610 17-649                      |
| N\$SNOD | 000012   | 8-178 11-305                                                     |
| N\$SACC | = 000001 | 40-1606                                                          |
| N\$SBUF | = *****  | 22-896                                                           |
| N\$SECL | = *****  | 8-206 10-263 11-326 17-706                                       |
| N\$SEVL | = 000001 | #4-2 11-317 11-326                                               |
| N\$SHDR | = 000007 | 27-1115                                                          |
| N\$MCP  | = *****  | 30-1225                                                          |
| N\$SMLL | = 000001 | 8-129 8-196 10-259 11-347 19-786 31-1255 35-1387 35-1423 42-1711 |
| N\$SVCT | = 000001 | 8-206 10-263 17-706                                              |
| N\$SSES | = 000001 | #6-65 8-206 10-263 11-326 17-706                                 |
| N\$SLI  | = *****  | 8-158 35-1393 41-1669 42-1701                                    |
| N\$SVCT | = *****  | 8-131 8-198 9-238 10-259 10-266 11-298 11-347 17-609 17-660      |
|         |          | 17-664 17-720 19-780 19-786 20-806 21-850 21-851 21-853 23-929   |

\*\*FILE\*\*ID\*\*SESTIM

```

SSSSSSSS EEEEEEEF SSS SSSS TTTT TTTT I I I I I MM MM
SSSSSSSS EEEEEEEF SSS SSSS TTTT TTTT I I I I I MM MM
SS EE SS TT II MMMM MMMM
SS EE SS TT II MMMM MMMM
SS EE SS TT II MM MM MM
SS EE SS TT II MM MM
SSSSSSS EEEEEEEF SSSSSS TT II MM MM
SSSSSSS EEEEEEEF SSSSSS TT II MM MM
SS EE SS TT II MM MM
SS EE SS TT II MM MM
SS EE SS TT II MM MM
SSSSSSS EEEEEEEF SSSSSS TT I I I I I MM MM
SSSSSSS EEEEEEEF SSSSSS TT I I I I I MM MM

```

```

....
....
....
....

```

```

11 11 SSSSSSS
11 11 SSSSSSS
1111 1111 SS
1111 1111 SS
11 11 SS
11 11 SS
11 11 SSSSSS
11 11 SSSSSS
11 11 SS
11 11 SS
11 11 SS
11 11 SS
111111 111111 SSSSSSS
111111 111111 SSSSSSS

```



```

520 .IF DF N$$MCP
521
522 .SBTTL Map prototype TCB into target TCB
523
524 ;+
525 ;**MPTCB-Map prototype TCB into target TCB
526
527 Generate a new TCB for the given task and link it into the
528 STD.
529
530 Inputs:
531 R0 = Virtual address of prototype TCB
532 R3 = Pointer to RAD50 task name
533 R4 = Address of CCB
534
535 Outputs:
536 R0 = Address of new TCB
537 'c' Clear - New TCB linked into STD
538 'c' Set - Failed to allocate a new TCB
539 APR 6 mapping is restored to netacp
540
541 Registers modified:
542 R0, R1
543
544 .PSECT
545
546 MPTCB: SAVRG <R2,R4>
547 MOV T.NAM+2(R0),-(SP) ; Put task name on the stack
548 MOV T.NAM(R0),-(SP) ; ...
549 RECMAP ; Recover APR 6
550 JMP TCB ; Change to run in APR 6
551
552 .PSECT $HIGH
553
554 TCB: CLR R1 ; Initialise count
555
556 .IF DF R$$MPL
557
558 BITB #CX.SMC,C.MOD(R4)
559 BEQ 20$; If EQ, non-multi-copy task, just try once
560
561 .ENDC
562
563 10$: CALL MUTNAM ; Generate a new name
564 20$: MOV SP,R3 ; Point to new task name
565 CALL @SRSTD ; See if name is already in use
566 RCS 40$; If CS, it is not
567 BIT #T3.REM,T.ST3(R0)
568 BNE 30$; If marked for removal we cannot use it
569 BIT #TS.EXE,T.STAT(R0)
570 BNE 50$; If not active, use it
571
572 30$: INC R1 ; Update task #
573 CMP R1,$RQCPY ; Checked all possible task names?
574 BNE 10$; If NE, no ... keep trying
575 BR 60$; All done
576
577 40$: MOV TLGTH,R1 ; Get size of a TCB

```

```

42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,EVT$,CALLX,MAPLLT,RECMAP
45 .MCALL DHBDF$,PDVDF$,EVLDF$,ECDDB$,CTRDF$,MSGDF$
46
47 000000 DHBDF$; Define DEC home block offsets
48 000000 PDVDF$; Define PDV offsets
49 000000 EVLDF$; Define event logger constants
50 000000 ECDDB$; Define ECL database offsets
51 000000 CTRDF$; Define counter block offsets
52 000000 MSGDF$; Define message constants
53
54 051516 SES$DV = 'NS ; Device name for session control
55
56 000001 N$$SES = 1 ; This module is part of session control

```

SEINI11S CREATED BY MACRO ON 28-JUN-85 AT 19:56 PAGE 7 C 3

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES  |
|---------|------------|-------------|
| \$SESDB | = ***** GX | *7-82 8-157 |
| \$SESPD | = ***** GX | *7-78       |
| \$UCB   | = ***** GX | *7-94 7-114 |
| \$XPTIN | = ***** GX | 7-106       |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                                                                         |
|---------|------------|------------------------------------------------------------------------------------|
| ACPIDL  | = ***** GX | 8-93                                                                               |
| AC\$DNT | = 000002   | #6-54                                                                              |
| AC\$X25 | = 000001   | #6-54                                                                              |
| AE\$CIR | = 000003   | #6-54                                                                              |
| AE\$LIN | = 000001   | #6-54                                                                              |
| AE\$MOD | = 000004   | #6-54                                                                              |
| BYTE3   | = 000300   | #8-118 8-121 #8-118 8-121 #8-118 8-121 #8-118 8-121 8-118 8-121 8-118 8-121 #8-121 |
| CL\$ASZ | = 010500   | #6-54                                                                              |
| CL\$DLL | = 000500   | #6-54 6-54 6-54 6-54 6-54 6-54 6-54 6-54 6-54 6-54                                 |
| CL\$ECL | = 000300   | #6-54                                                                              |
| CL\$LDN | = 010400   | #6-54                                                                              |
| CL\$MAN | = 000000   | #6-54 6-54 6-54                                                                    |
| CL\$PAZ | = 034100   | #6-54 6-54 6-54                                                                    |
| CL\$PLH | = 034000   | #6-54 6-54 6-54                                                                    |
| CL\$PLL | = 000600   | #6-54                                                                              |
| CL\$PRT | = 034200   | #6-54                                                                              |
| CL\$ROU | = 010000   | #6-54 6-54 6-54                                                                    |
| CL\$SES | = 000200   | #6-54 6-54 6-54                                                                    |
| CL\$SGE | = 035000   | #6-54 6-54 6-54                                                                    |
| CL\$SSE | = 035100   | #6-54 6-54 6-54 6-54 6-54 6-54 6-54 6-54                                           |
| CL\$TRN | = 000400   | #6-54 6-54 6-54 6-54 6-54 6-54 6-54 6-54 6-54 6-54                                 |
| CL\$XL2 | = 013700   | #6-54                                                                              |
| CL\$XL3 | = 013600   | #6-54                                                                              |
| CL\$X25 | = 013500   | #6-54 6-54 6-54                                                                    |
| DEALDB  | = ***** GX | 8-113                                                                              |
| DL\$AST | = 000002   | #6-54                                                                              |
| DL\$HLT | = 000000   | #6-54                                                                              |
| DL\$IST | = 000001   | #6-54                                                                              |
| DL\$MAI | = 000004   | #6-54                                                                              |
| DL\$OFF | = 000001   | #6-54                                                                              |
| DL\$ON  | = 000000   | #6-54                                                                              |
| DL\$RUN | = 000003   | #6-54                                                                              |
| DL\$SHU | = 000002   | #6-54                                                                              |
| DL\$SYN | = 000005   | #6-54                                                                              |
| DREXT   | = ***** GX | 8-124                                                                              |
| EF\$ACT | = 000001   | #6-54                                                                              |
| EV\$SES | = ***** GX | 8-118                                                                              |
| EV\$ACF | = 000201   | #6-54                                                                              |
| EV\$ADR | = 000420   | #6-54                                                                              |
| EV\$ADU | = 000417   | #6-54                                                                              |
| EV\$APL | = 000400   | #6-54                                                                              |
| EV\$ARC | = 000421   | #6-54                                                                              |
| EV\$AUC | = 000010   | #6-54                                                                              |
| EV\$AUS | = 000003   | #6-54                                                                              |
| EV\$CDF | = 000520   | #6-54                                                                              |
| EV\$COZ | = 000011   | #6-54                                                                              |
| EV\$DBR | = 000302   | #6-54                                                                              |
| LV\$GAC | = 035101   | #6-54                                                                              |

```

193 MOVWB C.LIN(R4),L.CHN(R3) ; Save channel # for possible DC response
194
195 .ENDC
196
197 000302 COUNT$ E$NMR ; Count a received ECL message
198 000310 COUNT$ E$NCR ; Count a received connect
199
200 000316 CALL PROCON ; Process requested connect services
201 000322 CALL CNVCI ; Process descriptor fields
202 000326 103417 BCS 110$; If CS, field format error or resource error
203
204 000330 MOVWB #ST$CIR,(R3) ; Indicate CI received
205 000334 016564 000030 000006 MOV N$PLLT(R5),C.LIN(R4)
206 000342 CALL USRCI ; Tell user about received CI message
207 000346 103405 BCS 100$; If CS, connect failure
208
209 000350 40$: .IF DF N$SLI
210
211 BIT #LT.SLI*400,(R3); Is this a system level interface link?
212 BNE 50$; If NE, yes ... don't send the CI ACK yet
213
214 .ENDC
215
216 000350 CALLE ACKCI ; Send the CI ACK
217 000360 50$: RETURN
218
219 ;+
220 ; Error processing for CI message
221 ;-
222 000362 100$: CALL RLSCI ; Release the CI resources
223
224 000366 022765 000001 000022 110$: CMP #ER$RES,N$ERRC(R5) ; Is this a 'NO RESOURCES'
225 000374 001414 BEQ 115$; If EQ, yes - send a DC message
226 000376 016746 000000G MOV $RCCB,-(SP) ; Save the RDB so we can toss it later
227 000402 112713 000012 MOVWB #ST$DIP,(R3) ; Put link in DIP state
228 000406 016563 000022 000100 MOV N$ERRC(R5),L.DCR(R3) ; Set up the correct reason code
229 000414 CALL SENDDI ; Send the DI message
230 000420 012667 000000G MOV (SP)+,$RCCB ; Recover the CI message RDB
231 000424 000414 BR 130$; and return so the CI RDB is tossed
232
233 000426 115$: CALL REMLNK ; Remove the LLT
234 000432 005065 000020 CLR N$DLA(R5) ; Make sure we have the correct link addresses
235
236 000436 016704 000000G 120$: MOV $RCCB,R4 ; Recover received CCB address
237 000442 CALLE SENDDC ; Send DC for no resources, etc
238 000452 005067 000000G CLR $RCCB ; We have re-used this CCB
239 000456 130$: RETURN

```

|                 |                 |                  |                  |                   |
|-----------------|-----------------|------------------|------------------|-------------------|
| LA.ACK= 100000  | L.NXN 000016    | NF\$TIM= 000200  | PROCON 001352R   | T\$FLAG 000044    |
| LA.CRS= 020000  | L.NXTH 000010   | NM\$ARA= 176000  | P\$P45= 000000   | T\$IF 000013      |
| LA.MSK= 170000  | L.OPD 000103    | NM\$NOD= 001777  | P\$WRD= 000000   | T\$IFL 000013     |
| LA.NAK= 110000  | L.OPDL 000102   | NOLINK= ***** GX | Q\$SOP= 000010   | T\$IFO 000013     |
| LA.NMS= 010000  | L.REM 000006    | NOOP 000076R     | RCPTBL 000004R   | T\$IFS 000013     |
| LA.RES= 040000  | L.RFC 000050    | N\$SDON= 000000  | RCVCC 000552R    | T\$LIN 000000     |
| LA.WND= 004000  | L.RLA 000004    | N\$SDI= 000002   | RCVCI 000120R    | T\$LIP 000006     |
| LD\$LP= 000000  | L.RNO 000022    | N\$SWDC= 000004  | RCVCL 000000R    | T\$LLD 000012     |
| LF.DRD= 000004  | L.RTO 000060    | NT\$AKD= 000020  | RCVDC 001176R    | T\$LLDC 000045    |
| LF.FRC= 000001  | L.RTYD 000055   | NT\$AKI= 000022  | RCVDI 001022R    | T\$LLDL 000012    |
| LF.HFO= 000010  | L.RTYI 000057   | NT\$CC= 000016   | RCVRCI 000460R   | T\$LLDO 000012    |
| LF.HMF= 000040  | L.SEC 000064    | NT\$CON= 000000  | RCVTBL 000100R   | T\$LLDS 000012    |
| LF.HSF= 000020  | L.SEGZ 000076   | NT\$CTL= 000000  | REMLNK= ***** GX | T\$LEN 000046     |
| LF.IRD= 000002  | L.STA 000000    | NT\$DAT= 000002  | RF.INI= 000040   | T\$LOPS 000002    |
| LF.MMF= 000200  | L.TC 000042     | NT\$DC= 000012   | RF.LDA= 000004   | T\$TCL 000024     |
| LF.MSF= 000100  | L.TIC 000043    | NT\$DIS= 000014  | RF.ROR= 000010   | T\$TIM 000026     |
| LS.DLS= 100000  | L.TIPD 000013   | NT\$DLS= 000006  | RF.RSV= 000100   | T\$TPR 000014     |
| LS.FCC= 000004  | L.TIPI 000012   | NT\$ILS= 000010  | RF.RTS= 000020   | T\$TPT 000020     |
| LS.FCO= 000001  | L.TMRD 000054   | NT\$IMS= 000102  | RH.CTF= 000001   | T\$NAPL 000004    |
| LS.FCI= 000002  | L.TMRI 000056   | NT\$INT= 000004  | RH.DTF= 000002   | T\$NFE 000000     |
| LS.ILS= 100000  | L.TYP 000001    | NT\$RET= 000032  | RH.MSF= 000003   | T\$NLEN 000010    |
| LS.MAK= 000020  | L.USA 000024    | NT\$ROU= 000024  | RLSCI= ***** GX  | T\$NNUL 000002    |
| LS.MNK= 000040  | L.USTA 000036   | NT\$RTR= 000030  | RL.CKS= 000002   | T\$NOPL 000006    |
| LS.RFS= 000360  | L.VER 000015    | NT\$TSP= 000026  | RL.CTL= 000003   | T\$NPN 000042     |
| LS.RSV= 000300  | L.WND 000040    | N\$ACQ 000000    | RL.DAH= 000006   | T\$NPL 000005     |
| LT.CCA= 000020  | MA.CI= 000040   | N\$ACTL 000032   | RL.DH4= 000025   | T\$NRUL 000007    |
| LT.DIR= 000010  | MA.DA= 000000   | N\$CIR 000034    | RL.SGH= 000004   | T\$NVR 000001     |
| LT.LCL= 000001  | MA.LL= 000020   | N\$DLA 000020    | RL.TYP= 000012   | T\$RPRI 000040    |
| LT.LPL= 000002  | MC.CC= 000040   | N\$DLY 000014    | RL.VER= 000006   | T\$SVC 000034     |
| LT.NOT= 000040  | MC.CI= 000020   | N\$ELEN 000054   | RSNDDI 001654RG  | T\$T5 000030      |
| LT.RSU= 000200  | MC.DC= 000100   | N\$ENC 000042    | RTRANS 001672RG  | T\$T6 000032      |
| LT.SLI= 000004  | MC.DI= 000060   | N\$ERRC 000022   | R\$SDER= 000000  | T\$KMG= 000000    |
| LT.TDA= 000100  | MC.NO= 000000   | N\$FLG 000005    | R\$K71= 000001   | T\$MIN= 000000    |
| L\$BASG= 000000 | MC.RC= 000140   | N\$FNC 000006    | R\$SND= 000000   | USRCC= ***** GX   |
| L\$DRV= 000000  | MD.BM= 000040   | N\$GENQ 000052   | R\$S11M= 000000  | USRCI= ***** GX   |
| L\$P11= 000001  | MD.EM= 000100   | N\$GTM 000015    | R\$S11S= 000000  | USRINT= ***** GX  |
| L\$11R= 000000  | MD.ILS= 000040  | N\$HIGH 000033   | SAVOPT= ***** GX | US\$CNF= 000002   |
| L.CSTA 000037   | MD.IM= 000020   | N\$LLT 000026    | SENDDC= ***** GX | US\$DIS= 000006   |
| L.CTR 000074    | MF.ACK= 000004  | N\$LLTM 000024   | SENDDI 001552RG  | US\$DON= 000000   |
| L.DCR 000100    | MF.CTL= 000010  | N\$LVC 000036    | SNSESD= ***** GX | US\$DSC= 000004   |
| L.FLAG 000014   | MF.DAT= 000000  | N\$MBXQ 000050   | STOPCC= ***** GX | US\$EAC= 000012   |
| L.ILSO 000052   | M\$CRB= 000124  | N\$PLLT 000030   | STOPCI= ***** GX | US\$WDS= 000010   |
| L.ILTT 000066   | M\$CRX= 000000  | N\$SLA 000016    | STCC= 000004     | V\$CIR= 001000    |
| L.LDA 000032    | M\$FCS= 000000  | N\$SNOD 000012   | STCIR= 000006    | X\$DBT= 000000    |
| L.LIA 000034    | M\$MGE= 000000  | N\$TIM 000004    | STCIS= 000002    | \$BYTE= ***** GX  |
| L.LLG 000002    | M\$MUP= 000000  | N\$VCB 000010    | STDAT= 000010    | \$CALLX= ***** GX |
| L.LNO 000026    | M\$NET= 000000  | N\$ACC= 000001   | STDIP= 000012    | \$INFO= ***** GX  |
| L.LPT 000065    | M\$OVR= 000000  | N\$ACK= 000011   | STSPND= 000014   | \$OPDAT= ***** GX |
| L.LSA 000030    | NC.FM0= 000000  | N\$EVL= 000001   | SS\$WRG= 000000  | \$JPLNG= ***** GX |
| L.LSFD 000046   | NC.FM1= 000001  | N\$HDR= 000007   | SS\$YSZ= 000600  | \$TCB= ***** GX   |
| L.LSFI 000044   | NC.FM2= 000002  | N\$LDV= 000001   | TI.END= 000003   | \$REASN= ***** GX |
| L.LTT 000062    | NF\$BLK= 000100 | N\$MLL= 000001   | TI.RSV= 000370   | \$SEGMT= ***** GX |
| L.MASQ 000070   | NF\$DMO= 000010 | N\$MOV= 000010   | TI.RTI= 000002   | \$SRVCS= ***** GX |
| L.MAST 000073   | NF\$MOU= 000040 | N\$NCT= 000001   | TI.RT2= 000001   | \$SHIT= 000004    |
| L.MASZ 000072   | NF\$RST= 000002 | N\$OVR= 000022   | TI.TYP= 000003   | \$ERCP 000000RG   |
| L.NIN 000020    | NF\$SCN= 000020 | N\$PEM= 000001   | TI.VFY= 000004   | \$\$\$\$= 000034  |
|                 | NF\$SHU= 000004 | N\$SES= 000001   |                  |                   |

```

130 .SBITL Connect request completion
131
132 +
133 **--USRFC-Connect request completion
134 A connect request has been completed with a connect confirm (CC)
135 message.
136 -
137 Inputs:
138 R3 = Virtual address of LLT
139 R5 = Address of database descriptor
140
141 Registers modified:
142 R0, R1, R2
143
144 000136 USRCC:: CALL CPYCNC ; Copy optional data
145 000142 012000 000000C MOV #IS.SUC&377,R0 ; Assume successful completion
146 000146 103002 BCC 10$; If CC, yes
147 000150 012700 000000C MOV #IS.DA0&377,R0 ; else data overrun
148
149 000154 016302 000040 10$: MOV L.WIND(R3),R2 ; Get address of window block
150
151 000160 016362 000076 000006 MOV L.SEGZ(R3),W.SEGZ(R2)
152 000166 016267 000010 000000G MOV W.TMP(R2),$IOPKI ; Set up address of I/O packet
153 000174 005062 000010 CLR W.TMP(R2) ; No I/O packet waiting now
154
155 000200 CALL IODUN ; Perform I/O completion
156
157 000204 016503 000026 MOV N$LLT(R5),R3 ; Recover LLT address
158 000210 MAPLLT ; and mapping
159 000216 RETURN

```

SESSL1 - Session control system MACRO V05.03b Friday 28-Jun-85 19:58<sup>D.8</sup>  
Table of contents

6- 42 Macro definitions



```

415 .SBTTL Get local node information
416
417 ;+
418 ;*-SLIGLN-Get local node information
419 Copy information about the local node to the buffer supplied.
420
421 Inputs:
422 R4 = Address of CCB
423 C.BUF1 - Buffer descriptor
424 C.FLG2+1 - Information type
425 C.STA - Source PDV index
426 R5 = Address of database descriptor
427
428 SLIGLN: MOV C.BUF+2(R4),R2 ; Get pointer to buffer
429 MOV C.FLG2+1(R4),R3 ; Get type of information required
430 ASL R3 ; Form word index
431 MOV #WORK,R0 ; Point to workspace area
432 MOV @DECPT,R1 ; Point to the DEC home block
433
434 CALL @GLNTBL(R3) ; Get information
435 MOV #S.EUNN,R1 ; Assume information not available
436 BCS 100$; If CS, info not available
437 MOV #S.SSUC,R1 ; Successful completion
438 SUB #WORK,R0 ; Compute length of return data
439
440 CMP R0,C.CNT(R4) ; Did user supply enough buffer space
441 BLOS 10$; If LOS, yes
442 MOV C.CNT(R4),R0 ; Use user supplied buffer size
443 MOV #S.ELST,R1 ; Indicate data lost
444
445 10$: MAP C.BUF(R4) ; Map to the user buffer
446 MOV #WORK,R3 ; Point to the returned results
447 20$: MOV (R3)+(R2)+ ; Copy data to supplied buffer
448 SOB R0,20$; ...
449
450 100$: CALLR XMECMP ; Complete the request
451
452 ;+
453 ; Get local node information dispatch table
454 ;*-
455 .PSECT $HIGH
456
457 GLNTBL: .WORD GLNLDC ; Get local node info
458 .WORD GLNADD ; Get local node address
459 .WORD GLNMAP ; Get node address/name mapping
460 .WORD GLNHOS ; Get host node address

```

```

846 .SBTTL Move optional data to the internal buffer
847
848 ;+
849 ;**--MOVDAT--Move optional data to the internal buffer
850 ;
851 ; Copy the optional data (if any) to the internal optional data buffer.
852 ;
853 ; Inputs:
854 ; R3 = Virtual address of LLT
855 ; R4 = Address of CCB
856 ; C.BUF1 - Optional data descriptor
857 ; R5 = Address of database descriptor
858 ;
859 ; Registers modified:
860 ; R0, R1, R2
861 MOVDAT: MOV C.CNT(R4),R0 ; Get # of bytes to be moved
862 MOV R0,$OPLNG ; and into local count
863 BEQ 20$; If EQ, none
864
865 MAP C.BUF(R4) ; Map to the buffer
866 MOV C.BUF+2(R4),R1 ; Get buffer's virtual address
867 MOV #$OPDAT,R2 ; Point to the internal buffer
868 10$: MOV (R1)+,(R2)+ ; Copy the optional data
869 SOB R0,10$; ...
870 MAPLLT ; Restore mapping to the LLT
871
872 20$: RETURN
873
874 .ENDC
875
876 .END

```

000001

```

211 .SBTTL Add mail to mailbox
212
213 ;*--ADDMAI-Add mail to mailbox
214
215 Add a piece of mail to the end of a mailbox queue.
216
217 Inputs:
218 R4 = Address of CCB
219 R5 = Address of database descriptor
220
221 Registers modified:
222 R0, R1
223
224 ADDMAI::SAVRG <R3> ; Get a free register
225 000360 016703 000000G MOV $MAIBX,R3 ; Get address of mailbox to use
226
227 000364 005014 CLR (R4) ; Clear the Link word
228 000366 010300 MOV R3,R0 ; Compute address of mailbox listhead
229 000370 062700 000014 ADD #M.MAIL,R0 ; ...
230
231 000374 010001 10$: MOV R0,R1 ; Save current position in list
232 000376 011000 MOV (R0),R0 ; Get next queue entry
233 000400 001375 BNE 10$; Scan to the end of the list
234 000402 010411 MOV R4,(R1) ; Link CCB at the end of the list
235
236 000404 005763 000012 TST M.SPA(R3) ; Any AST service required?
237 000410 001404 BEQ 20$; If EQ, no
238 000412 CALLE SPRAST ; Spring the AST
239
240 000422 20$: RESRG <R3> ; Restore register
241 000424 RETURN

```

|     |        |        |                |               |                   |  |                                              |
|-----|--------|--------|----------------|---------------|-------------------|--|----------------------------------------------|
| 700 | 001560 | 001433 |                | BEQ           | 130\$             |  | ; If EQ, no                                  |
| 701 |        |        |                |               |                   |  |                                              |
| 702 | 001562 | 005067 | 000000G        | CLR           | \$BYTE            |  | ; Set up byte count                          |
| 703 | 001566 | 111267 | 000000G        | MOV           | (R2), \$BYTE      |  |                                              |
| 704 | 001572 | 066764 | 000000G 000020 | ADD           | \$BYTE, C.CNT(R4) |  | ; Update byte count for user                 |
| 705 | 001600 | 016705 | 000000G        | MOV           | \$SESD, R5        |  | ; Recover address of ECL database descriptor |
| 706 | 001604 |        |                | COUNT         | \$E\$NBR          |  | ; Count ECL bytes received                   |
| 707 |        |        |                |               |                   |  |                                              |
| 708 | 001612 | 004367 | 000044         | JSR           | R3, COPYIMG       |  | ; Copy image field                           |
| 709 | 001616 | 140    |                | .BYTE         | N.CDAC            |  | ; Optional user data                         |
| 710 | 001617 | 020    |                | .BYTE         | 16                |  | ; Max length = 16 bytes                      |
| 711 | 001620 | 103013 |                | BCC           | 130\$             |  | ; If CC, valid field                         |
| 712 |        |        |                |               |                   |  |                                              |
| 713 | 001622 | 016403 | 000016         | 100\$: MOV    | C.BUF+2(R4), R0   |  | ; Recover pending connect block address      |
| 714 | 001626 | 162700 | 000004         | SUB           | #4, R0            |  | ; Backup over task name                      |
| 715 | 001632 | 012701 | 000166         | MOV           | #N.CBL+16.+4, R1  |  | ; Size of the block                          |
| 716 | 001636 |        |                | CALL          | @DEACB            |  | ; Deallocate the block                       |
| 717 |        |        |                |               |                   |  |                                              |
| 718 | 001642 |        |                | 110\$: CALL   | @CCBRT            |  | ; Release the CCB                            |
| 719 | 001646 | 000261 |                | 120\$: SEC    |                   |  | ; Indicate error                             |
| 720 | 001650 |        |                | 130\$: RESMAP |                   |  | ; Restore mapping                            |
| 721 | 001654 |        |                | RESRG         | <R3, R5>          |  | ; and registers                              |
| 722 | 001660 |        |                | RETURN        |                   |  |                                              |

```

1159 .SBTTL Kill logical link
1160 ;+
1161 ;**KILLNK-Kill logical link
1162 ;
1163 ; Kill logical by setting the major state to disconnect in progress
1164 ; and setting the user and network substates according to the
1165 ; supplied state table.
1166 ;
1167 ; Inputs:
1168 ; R2 = Address of substate table
1169 ; R3 = Virtual address of LLT
1170 ; R5 = Address of database descriptor
1171 ;
1172 ; Registers modified:
1173 ; R0, R2
1174 ;
1175 002364 .PSECT
1176
1177 002364 111300 KILLNK::MOVB (R3),R0 ; Get current major link state
1178 002366 060002 ADD R0,R2 ; Compute address of table entry
1179 002370 021227 177777 CMP (R2),#-1 ; Is there a change in user sub state ?
1180 002374 001414 BEQ 30$; If EQ, no
1181 002376 112713 000012 MOVB #STDIP,(R3) ; Else, set state to disconnect in progress
1182 002402 112200 MOVB (R2)+,R0 ; Get user substate
1183 002404 100402 BMI 10$; Is M1, leave user state alone
1184 002406 110063 000036 MOVB R0,L.USTA(R3) ; Set user substate
1185
1186 002412 112200 10$: MOVB (R2)+,R0 ; Get network substate
1187 002414 100402 BMI 20$; If M1, leave network substate alone
1188 002416 110063 000037 MOVB R0,L.CSTA(R3) ; Set network substate
1189
1190 002422 20$: CALL TRMLNK ; Try to terminate the link
1191 002426 30$: RETURN

```

```

1620 .SBTTL Terminate a logical link
1621
1622 *--TRMLNK-Terminate a logical link
1623 *--TRMNET-Terminate network side of logical link
1624
1625 Attempt to complete the termination of a logical link.
1626
1627 Inputs:
1628 R3 = Virtual address of LLT
1629 R5 = Address of database descriptor
1630
1631 Registers modified:
1632 R0, R1, R2, R3, R4
1633
1634 003450 TRMLNK::CALL TRMUSR ; Terminate user side of logical link
1635
1636 003454 116300 000037 TRMNET::MOVB L,CSTA(R3),R0 ; Get disconnect state for network
1637 003460 CALL @NSTATE(R0) ; Process network disconnect state
1638
1639 003464 005763 000012 TST L,TIPI(R3) ; Any outstanding transmits?
1640 003470 001006 BNE 10$; If NE, yes ... can't terminate yet
1641
1642 003472 026327 000036 000000 CMP L,USTA(R3),#<N$DON*400>+US$DON
1643 003500 001002 BNE 10$; If NE, disconnect not yet complete
1644
1645 003502 CALL REMLNK ; Remove the link database
1646 003506 10$:
1647 003506 NOOP:
1648 003506 DONE: RETURN
1649
1650 ;+
1651 ; Disconnect table for network
1652 ;+
1653
1654 003510 003506* NSTATE::WORD DONE ; Processing complete
1655 003512 000000G .WORD SENDDI ; Send disconnect initiate message
1656 003514 000000G .WORD RSNDI ; Retransmit disconnect initiate message
1657
1658 ;+
1659 ; Disconnect state for user
1660 ;+
1661
1662 003516 003506* USTATE::WORD DONE ; Processing complete
1663 003520 000000G .WORD USRCNF ; Connect failed
1664 003522 000000G .WORD USRDSC ; User disconnect complete
1665 003524 000000G .WORD USRDIS ; Notify user of disconnect or abort
1666 003526 003506* .WORD NOOP ; Wait for disconnect acknowledgement
1667 003530 000000G .WORD USRDIS ; Error response to connect accept
1668
1669 .IF DF N$SLI
1670
1671 ;+
1672 ; Disconnect state for user (system level interface)
1673 ;+
1674
1675 SSTATE::WORD DONE ; Processing complete
1676 .WORD SLICNF ; Connect failed
1677

```

SESSUB11S CREATED BY MACRO ON 28-JUN-85 AT 20:00 PAGE 7 C 15  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE    | REFERENCES | 23-951   | 23-956   | 27-1108  | 30-1211  | 30-1237 | 34-1363 | 34-1365 | 34-1368 | 35-1387 |
|---------|----------|------------|----------|----------|----------|----------|---------|---------|---------|---------|---------|
| N.CACC  | 000116   |            | 23-951   | 23-956   | 27-1108  | 30-1211  |         |         |         |         |         |
| N.CBL   | = 000142 |            | 35-1404  | 35-1425  | 40-1602  | 42-1711  |         |         |         |         |         |
| N.CDAC  | 000140   |            | 17-694   |          |          |          |         |         |         |         |         |
| N.CIDC  | 000062   |            | 17-626   | 17-698   | 17-715   | 37-1485  |         |         |         |         |         |
| N.CPSC  | 000104   |            | 17-709   |          |          |          |         |         |         |         |         |
| N.SFM   | 000036   |            | 17-684   |          |          |          |         |         |         |         |         |
| N.SND   | 000030   |            | 17-689   |          |          |          |         |         |         |         |         |
| OBJHD   | = *****  | GX         | 17-671   | *17-655  | *17-656  | *17-657  |         |         |         |         |         |
| OP\$INI | = 000000 |            | 15-468   |          |          |          |         |         |         |         |         |
| OP\$TER | = 000001 |            | 30-1212  |          |          |          |         |         |         |         |         |
| O.FLG   | 000003   |            | #6-54    |          |          |          |         |         |         |         |         |
| O.NAM   | 000006   |            | #6-54    |          |          |          |         |         |         |         |         |
| O.TYP   | 000002   |            | 30-1233  |          |          |          |         |         |         |         |         |
| PH\$HDE | = 000004 |            | 30-1231  | 30-1232  |          |          |         |         |         |         |         |
| PH\$LOC | = 000002 |            | 30-1219  |          |          |          |         |         |         |         |         |
| PH\$MTS | = 000003 |            | #6-54    |          |          |          |         |         |         |         |         |
| PH\$UMP | = 000000 |            | #6-54    |          |          |          |         |         |         |         |         |
| PH\$WCS | = 000001 |            | #6-54    |          |          |          |         |         |         |         |         |
| PNTCCB  | 002516   | RG         | #31-1255 |          |          |          |         |         |         |         |         |
| PROIMG  | 002554   | R          | 18-752   | #32-1291 | 33-1344  |          |         |         |         |         |         |
| PRONAM  | 002576   | R          | 17-667   | 17-672   | #33-1321 |          |         |         |         |         |         |
| P.HDR   | = *****  | GX         | 38-1509  |          |          |          |         |         |         |         |         |
| REJECT  | 002670   | RG         | 24-1005  | #34-1362 |          |          |         |         |         |         |         |
| REMLNK  | 002742   | RG         | #35-1387 | 41-1645  |          |          |         |         |         |         |         |
| RETRES  | 003074   | RG         | 23-973   | 35-1391  | 35-1409  | #36-1446 |         |         |         |         |         |
| RETTB   | 003134   | R          | 36-1454  | #36-1464 |          |          |         |         |         |         |         |
| RE\$ADC | = 000004 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$ADF | = 000017 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$ADR | = 000007 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$BLK | = 000010 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$CAF | = 000014 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$DAT | = 000001 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$DRP | = 000016 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$LDT | = 000013 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$LSN | = 000012 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$NML | = 000001 |            | #6-54    |          |          |          |         |         |         |         |         |
| PE\$OPE | = 000004 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$OPR | = 000000 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$RCV | = 000001 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$SED | = 000011 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$SKW | = 000006 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$STA | = 000002 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$SUM | = 000003 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$SYN | = 000000 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$TME | = 000021 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$TMO | = 000000 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$TMR | = 000020 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$UPT | = 000002 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$URE | = 000003 |            | #6-54    |          |          |          |         |         |         |         |         |
| RE\$VER | = 000005 |            | #6-54    |          |          |          |         |         |         |         |         |

SESTIM - Session control timer MACRO V05.03b Friday 28-Jun-85 20:01 <sup>C 16</sup>  
Table of contents

|    |     |                                               |
|----|-----|-----------------------------------------------|
| 6- | 42  | Macro definitions                             |
| 7- | 53  | Session control timer support                 |
| 8- | 109 | Session control timer service action routines |



|     |       |        |                   |                                 |
|-----|-------|--------|-------------------|---------------------------------|
| 577 |       | CALL   | @ALOCB            | ; Allocate a new TCB            |
| 578 |       | BCS    | 60\$              | ; If CS, allocation failure     |
| 579 |       |        |                   |                                 |
| 580 |       | MOV    | \$R0TCB,R1        | ; Recover prototype TCB address |
| 581 |       | CALL   | @TCBCP            | ; Copy TCB and link into STD    |
| 582 |       | BISB   | #CX.REM,C.MOD(R4) |                                 |
| 583 |       |        |                   |                                 |
| 584 | 50\$: | TST    | (PC)+             | ; Indicate success              |
| 585 | 60\$: | SEC    |                   | ; Indicate failure              |
| 586 |       |        |                   |                                 |
| 587 |       | BIT    | (SP)+,(SP)+       | ; Clean up stack                |
| 588 |       | RESRG  | <R4,R2>           |                                 |
| 589 |       | RETURN |                   |                                 |

```

58 .SBTTL Initialise session control ACP
59
60 +
61 **--SESINI--Initialise session control ACP
62 Perform all initialisation required for the session control ACP
63 at system state.
64
65 -
66 Outputs:
67 'C' Clear - Initialisation successful
68 'C' Set - Initialisation failed
69
70 Registers modified:
71 R0, R1, R2, R3, R4, R5
72
73 .PSECT $HIGH
74
75 000000 012702 017704 SESINI::MOV #*RECL,R2 ; Check if ECL process is loaded
76 000004 103527 CALL @PDVID ; Find PDV index
77
78 000012 110267 000000G MOV R2,$SESPD ; Save PDV index
79 000016 067702 000000G ADD @PDVTA,R2 ; Get pointer to PDV
80 000022 011202 MOV (R2),R2 ;
81 000024 016205 000016 MOV Z,DAT(R2),R5 ; Get address of session control database
82 000030 010567 000000G MOV R5,$SESD8 ; and save it
83 000034 005762 000012 TST Z,PCB(R2) ; Is the ECL process loaded?
84 000040 001513 BEQ 100$; If EQ, no
85
86 000042 016700 000000G MOV DEVHD,R0 ; Get device control block listhead
87 000046 011000 MOV (R0),R0 ; Get next DCB
88 000050 001507 BEQ 100$; If EQ, device not in system
89 000052 022760 051516 000000G CMP #SE$DV,D.NAM(R0)
90 000060 001372 BNE 10$; Keep looking for device
91 000062 005760 000000G TST D,UNIT(R0) ; Only unit zero?
92 000066 001367 BNE 10$; If NE, no
93
94 000070 016067 000000G 000000G MOV D,UCB(R0),$UCB ; Save UCB address for later
95
96 .IF DF N$SEXT
97
98 BITB #NFMOU,NFLG(R5)
99 BNE 20$; If NE, network is already mounted
100
101 .ENDC
102
103 000076 CALLX $ECLIN,ECL ; Initialise ECL process
104 000106 103470 BCS 100$; If CS, initialisation failed
105
106 000110 CALLX $XPTIN,XPT ; Initialise transport subsystem
107 000120 103463 BCS 100$; If CS, failed
108
109 000122 CALL ALOCDB ; Allocate databases
110 000126 103460 BCS 100$; If CS, failed
111
112 000130 EVT$ 2,0,...REOPR,SCON*400!SC$OFF
113
114 000144 016700 000000G MOV $UCB,R0 ; Recover UCB address

```

SESINI11S CREATED BY MACRO ON 28-JUN-85 AT 19:56 PAGE 8 D 3

MACRO CROSS REFERENCE

CREF 04.00

MACRO NAME REFERENCES

|         |       |       |       |       |       |
|---------|-------|-------|-------|-------|-------|
| CALL    | 7-75  | 7-09  | 8-161 | 8-174 | 9-196 |
| CALLX   | #6-44 | 7-103 | 7-106 |       |       |
| CTRDF\$ | #6-45 | 7-51  |       |       |       |
| DHBD\$  | #6-45 | 7-47  |       |       |       |
| ECDD\$  | #6-45 | 7-50  |       |       |       |
| EVLD\$  | #6-45 | 6-49  |       |       |       |
| EVT\$   | #6-44 | 7-112 |       |       |       |
| MAP     | #6-44 |       |       |       |       |
| MAPLLT  | #6-44 |       |       |       |       |
| MSGDF\$ | #6-45 | 6-52  |       |       |       |
| PDVDF\$ | #6-45 | 6-48  |       |       |       |
| RECMAP  | #6-44 |       |       |       |       |
| RESRG   | #6-44 |       |       |       |       |
| RETURN  | 7-140 | 8-172 | 8-176 | 9-198 |       |
| SAVRG   | #6-44 |       |       |       |       |
| SOB     | 8-166 |       |       |       |       |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE    | REFERENCES |
|----------|----------|------------|
| EV\$HCE  | = 035114 | #6-54      |
| EV\$HCI  | = 035113 | #6-54      |
| EV\$HFE  | = 000506 | #6-54      |
| EV\$IFL  | = 000413 | #6-54      |
| EV\$IFO  | = 000415 | #6-54      |
| EV\$IFS  | = 000414 | #6-54      |
| EV\$INF  | = 000515 | #6-54      |
| EV\$LDL  | = 000407 | #6-54      |
| EV\$LDN  | = 010416 | #6-54      |
| EV\$LDO  | = 000411 | #6-54      |
| EV\$LDS  | = 000410 | #6-54      |
| EV\$LSC  | = 000500 | #6-54      |
| EV\$LUP  | = 000412 | #6-54      |
| EV\$NOL  | = 000402 | #6-54      |
| EV\$NRC  | = 000416 | #6-54      |
| EV\$NSC  | = 000200 | #6-54      |
| EV\$NUL  | = 000401 | #6-54      |
| EV\$NVR  | = 000406 | #6-54      |
| EV\$OPL  | = 000403 | #6-54      |
| EV\$PCC  | = 034000 | #6-54      |
| EV\$PCI  | = 034001 | #6-54      |
| EV\$PCM  | = 034002 | #6-54      |
| EV\$PFE  | = 000404 | #6-54      |
| EV\$PPC  | = 034003 | #6-54      |
| EV\$RCF  | = 000517 | #6-54      |
| EV\$RDC  | = 010001 | #6-54      |
| EV\$RDR  | = 010002 | #6-54      |
| EV\$RJE  | = 035106 | #6-54      |
| EV\$RSC  | = 000501 | #6-54      |
| EV\$RUL  | = 000405 | #6-54      |
| EV\$SNA  | = 035000 | #6-54      |
| EV\$SNF  | = 000516 | #6-54      |
| EV\$SPE  | = 035001 | #6-54      |
| EV\$XCE  | = 034110 | #6-54      |
| EV\$XDI  | = 013600 | #6-54      |
| EV\$XGW  | = 034111 | #6-54      |
| EV\$XMX  | = 000514 | #6-54      |
| EV\$XRX  | = 000512 | #6-54      |
| EV\$XSC  | = 000513 | #6-54      |
| EV\$X2S  | = 013500 | #6-54      |
| EV.CCB   | = 000001 | #6-54      |
| EV.CIR   | = 000020 | #6-54      |
| EV.LCB   | = 000100 | #6-54      |
| EV.LIN   | = 000004 | #6-54      |
| EV.MAP   | = 000002 | #6-54      |
| EV.MOD   | = 000040 | #6-54      |
| EV.NOD   | = 000010 | #6-54      |
| EV.PRT   | = 000200 | #6-54      |
| EV\$DATA | = 000020 | #6-54      |
| EV\$EVT  | = 000000 | #6-54      |
| EV\$LCN  | = 000016 | #6-54      |
| EV\$LIN  | = 000000 | #6-54      |

```

241 .SBTTL Process a returned CI message
242 +
243 **--RCVRCI-Process a returned CI message
244 :
245 A connect initiate message has been returned to this node as
246 undeliverable, inform the user that the connect could not be
247 completed.
248 -
249 Inputs:
250 R2 = Pointer to field following source link address
251 R4 = Address of CCB
252 R5 = Address of database descriptor
253
254 000460 016500 000016 RCVRCI: MOV N$SLA(R5),R0 ; Get source link address for this message
255 000464 042700 177400 BIC #C<377>,R0 ; Isolate low byte as index
256 000470 026500 000036 CMP N$LVC(R5),R0 ; Is the link address within reason?
257 000474 103425 BLO 100$; If LO, no
258 000476 006300 ASL R0 ; Form word index
259 000500 066500 000040 ADD N$LVC+2(R5),R0 ; Compute address in logical link table
260 000504 011003 MOV (R0),R3 ; Get address of LLT
261 000506 001420 BEQ 100$; If EQ, the link does not exist
262
263 000510 016546 000016 MOV N$SLA(R5),-(SP) ; Save this links LLA
264 000514 ACCLT ; Gain access to the LLT
265 000524 022663 000002 CMP (SP)+,L.LLA(R3) ; Is this the link we're looking for?
266 000530 001007 BNE 100$; If NE, no
267 000532 122713 000002 CMPB #ST$CIS,(R3) ; Is the link in the correct state?
268 000536 001004 BNE 100$; If NE, no ... toss this message
269
270 000540 012701 000047 MOV #ER$COM,R1 ; Set up disconnect reason code
271 000544 CALL BRKLNK ; Break the logical link
272
273 000550 100$ RETURN

```

SESPRO - Session control protoc MACRO V05.03b Friday 28-Jun-85 19:57<sup>D 6</sup> Page 18-3  
Symbol table

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
001766 001 (RW,I,LCL,REL,CON)  
\$HIGH 000010 002 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 93  
Work file writes: 94  
Size of work file: 25334 Words ( 99 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:35.32  
SY:SESPRO11S.V2,[131,134]SESPRO11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCS/PA:1,[131,10]V2,SESPRO

```

161 .SBTTL Connect request failure
162
163 *
164 **--USRCNF-Connect request failure
165
166 A connect request failed because the CI message was returned from
167 an intermediate node or the CI request timed out.
168
169 -
170 Inputs:
171 R3 = Virtual address of LLT
172 R5 = Address of database descriptor
173
174 Registers modified:
175 R0, R1, R2
176
177 USRCNF::MOVB #US$DON,L.USTA(R3)
178 MOV $IOPKT,-(SP) ; Save any I/O packet
179 MOV #IE.NRJ&377,-(SP) ; Assume network reject
180 MOV L.DCR(R3),-(SP) ; Save reason for disconnect
181 BNE 10$; If NE, network disconnect
182 MOV #IE.URJ&377,2(SP)
183 CALL CPYCNC ; User disconnect, copy optional data
184 MOV R1,(SP) ; Save length of optional data
185 BCC 10$; If CC, successful
186 MOV #IE.DA0&377,2(SP)
187
188 10$: CALL RMVLNK ; Remove link resources
189
190 MOV (SP)+,R1 ; Get reason code for disconnect
191 MOV (SP)+,R0 ; Get network/user reject code
192 CALL IODUN ; Perform I/O completion
193 MOV (SP)+,$IOPKT ; Restore any I/O packet
194 RETURN

```

```

1 .TITLE SE3SLI - Session control system level interface
2 .IDENT /V05.00/
3 .ENABL LC
4
5 Copyright (C) 1982, 1983, 1985 by
6 Digital Equipment Corporation, Maynard, MASS.
7
8 This software is furnished under a license for use only on a
9 single computer system and may be copied only with the
10 inclusion of the above copyright notice. This software, or
11 any other copies thereof, may not be provided or otherwise
12 made available to any other person except for use on such
13 system and to one who agrees to these license terms. Title
14 to and ownership of the software shall at all times remain
15 in DEC.
16
17 The information in this document is subject to change without
18 notice and should not be construed as a commitment by Digital
19 Equipment Corporation.
20
21 DEC assumes no responsibility for the use or reliability of
22 its software on equipment which is not supplied by DEC.
23
24 Module description
25
26 Session control system level interface
27
28 Ident history:
29
30 4.00 07-NOV-83
31 DECnet-11M V4.0
32 DECnet-11M-PLUS V2.0
33
34 5.00 22-JUL-85
35 DECnet-11M/S V4.2
36 DECnet-11M-Plus V3.0
37 DECnet-Micro/RSX V1.0
38
39
40

```



```

462 .SBTTL Get local node info
463 ;+
464 ;**--GLNXXX-Get local node info
465 ;
466 Retrieve the required local node information to the workspace buffer.
467 ;
468 Inputs:
469 R0 = Address of workspace buffer
470 R1 = Address of DEC home block
471 R2 = Virtual address of user's buffer
472 R4 = Address of CCB
473 ;
474 Outputs:
475 R0 = Address of next available byte in workspace buffer
476 'C' Clear - Information available
477 'C' Set - Information not available
478 ;
479 Registers modified:
480 R1, R3
481 ;
482 ;+
483 ;**--GLNLOC-Get local node name
484 ;
485 GLNLOC: MOV R1,R3 ; Point to local node name
486 ADD #D$LNAM,R3 ; ...
487 ;
488 .REPT 3
489 MOV (R3)+,(R0)+ ; Copy local node name to buffer
490 .ENDR
491 ;
492 MOV D$SEG(R1),(R0)+ ; Copy local segment size into the buffer
493 CLC ; Indicate success
494 RETURN
495 ;
496 ;+
497 ;**--GLNADD-Get local node address and name
498 ;
499 GLNADD: MOV D$LNAM(R1),(R0)+ ; Fill in local node address
500 MOV R1,R3 ; Point to local node name
501 ADD #D$LNAM,R3 ; ...
502 CALL 'MODIMG ; Convert to image field
503 CLC ; Indicate success
504 RETURN
505 ;
506 ;+
507 ;**--GLNMAP-Get node address/name mapping
508 ;
509 ;
510 .SECT
511 ;
512 GLNMAP: MAP C.BUF(R4) ; Map to the supplied buffer
513 SAVRG <R2> ; Save registers
514 MOV #D$SNOD,R1 ; Point to alternate workspace
515 MOV #' ',(R1) ; and fill with spaces
516 MOV #' ',2(R1) ; ...
517 MOV #' ',4(R1) ; ...
518 MOV (R2)+,(R0)+ ; Get possible node address

```

|                |                 |                 |                  |                 |
|----------------|-----------------|-----------------|------------------|-----------------|
| ASSCHK= 000000 | CF.SOM= 000010  | C.BUF1 000014   | D\$RNN 000002    | FS.RLB= 002000  |
| ASSCPS= 000000 | CF.SYN= 000040  | C.BUF2 000024   | D\$RTMR 000076   | FS.RNG= 011000  |
| ASSPRI= 000000 | CF.TRN= 000100  | C.CNT 000020    | D\$SEG 000036    | FS.RST= 000000  |
| ASSTRP= 000000 | CL\$MFL= 000010 | C.CNT1 000020   | D\$SER 000032    | FS.RTN= 001000  |
| CB.ACN 000114  | CL\$SFL= 000004 | C.CNT2 000030   | D\$SRL 000052    | FS.SET= 005000  |
| CB.ACT 000112  | CL\$TYP= 000001 | C.FLG 000022    | D\$BUG= 177514   | FS.SFC= 005000  |
| CB.CCR= 000002 | CL.MU1= 000001  | C.FLG1 000022   | D\$ISK= 000000   | FS.SFR= 006000  |
| CB.DDM= 000040 | CL.MU2= 000002  | C.FLG2 000032   | D\$LL11= 000001  | FS.SFS= 004000  |
| CB.DFM 000006  | CL.RES= 177774  | C.FNC 000010    | D\$SYNC= 000000  | FS.SPW= 040000  |
| CB.DGR 000010  | CM.CIR= 000002  | C.LIN 000006    | D\$SYNM= 000000  | FS.STM= 000000  |
| CB.DLC= 000020 | CM.FMT= 100000  | C.LNK 000000    | ER\$ABM= 000010  | FS.STP= 002000  |
| CB.DL1 000010  | CM.HRD= 000002  | C.MOD 000011    | ER\$ABO= 000046  | FS.STR= 001000  |
| CB.DL2 000014  | CM.LIN= 000000  | C.NSP 000001    | ER\$ABT= 000011  | FS.TRM= 003000  |
| CB.DOB 000007  | CM.LOO= 000001  | C.PRO 000042    | ER\$ACC= 000042  | FS.WLB= 001000  |
| CB.DR1 000012  | CM.XLO= 000004  | C.RSV 000002    | ER\$CDI= 000052  | FS.XKL= 002000  |
| CB.DR2 000016  | CP.DCF= 000040  | C.STA 000007    | ER\$CDM= 000047  | FS.XOF= 010000  |
| CB.DUS 000012  | CP.HDL= 000007  | C.STS 000012    | ER\$CFMT= 000005 | FS.XON= 007000  |
| CB.LG1 000156  | CP.PS= 177400   | C.URM 177776    | ER\$MLB= 000006  | FS.ZER= 002000  |
| CB.NGD 000000  | CP.PSI= 000200  | C.XACP 000004   | ER\$NNF= 000012  | F\$LLVL= 000001 |
| CB.OPD 000134  | CP.XCF= 000100  | C.XID 000035    | ER\$NOD= 000002  | G\$STPP= 000000 |
| CB.OPT 000136  | CP.2FR= 000030  | C.XLEN 000044   | ER\$NSL= 000013  | G\$STSS= 000000 |
| CB.PSL 000100  | CS.ABO= 000100  | C.XPLI 000040   | ER\$NSR= 000003  | G\$STTK= 000000 |
| CB.PSW 000102  | CS.BRO= 000002  | C.XPT 000034    | ER\$RES= 000001  | G\$SWRD= 000000 |
| CB.RDB= 000004 | CS.BUF= 000200  | C.XSVC 000042   | ER\$STA= 000051  | IN.DAT= 000400  |
| CB.RDD 000056  | CS.CES= 000002  | C.XTC 000037    | ER\$UOB= 000004  | IN.ILS= 000001  |
| CB.RQ1 000060  | CS.CHN= 000010  | C.X25 000036    | ER\$XPR= 000000  | I\$RAR= 000000  |
| CB.SDB= 000010 | CS.CMP= 000200  | D\$AMXC 000072  | FL.CCP= 000020   | I\$SRDN= 000000 |
| CB.SFM 000032  | CS.DCR= 000400  | D\$AMXH 000074  | FC.CTL= 000006   | K\$CNT= 177546  |
| CB.SGR 000034  | CS.DEF= 000004  | D\$ANN 000000   | FC.KCP= 000016   | K\$CSR= 177546  |
| CB.SLI= 000100 | CS.DEV= 000002  | D\$BRPR 000102  | FC.KIL= 000004   | K\$SLDC= 000000 |
| CB.SL1 000034  | CS.DIS= 000040  | D\$BRTM 000100  | FC.MAN= 000024   | K\$STPS= 000074 |
| CB.SL2 000040  | CS.ENA= 000001  | D\$DELV 000045  | FC.MLD= 000026   | LA.ACK= 100000  |
| CB.SDB 000033  | CS.ENB= 000020  | D\$DELW 000046  | FC.PCT= 000030   | LA.CRS= 020000  |
| CB.SR1 000036  | CS.FRR= 100000  | D\$END = 000104 | FC.PWR= 000022   | LA.MSK= 170000  |
| CB.SR2 000042  | CS.I TL= 001000 | D\$FNB 000034   | FC.RCE= 000002   | LA.NAK= 110000  |
| CB.SUS 000036  | CS.HCR= 000001  | D\$HIOR 000024  | FC.RCP= 000014   | LA.NMS= 010000  |
| CB.XLB= 000001 | CS.HFE= 002000  | D\$HOST 000022  | FC.TIM= 000010   | LA.RES= 040000  |
| C.LLC= 000200  | CS.LST= 040000  | D\$INAC 000044  | FC.XCP= 000012   | LA.WND= 004000  |
| CE.ABO= 100362 | CS.MTL= 004000  | D\$INCT 000042  | FC.XME= 000000   | LD\$LP = 000000 |
| CE.DAO= 100346 | CS.RNG= 000010  | D\$IPL 000051   | FS.AST= 000000   | LF.DRD= 000004  |
| CE.DIS= 100366 | CS.ROV= 000004  | D\$LID 000020   | FS.CIB= 002000   | LF.FRC= 000001  |
| CE.ERR= 100370 | CS.RSN= 010000  | D\$LNAM 000006  | FS.CRA= 001000   | LF.HFO= 000010  |
| CE.ILN= 100350 | CS.SHU= 000001  | D\$LNUM 000014  | FS.DIS= 013000   | LF.HMF= 000040  |
| CE.LTO= 100356 | CS.SIP= 000302  | D\$LST 000047   | FS.DVC= 001000   | LF.HSF= 000020  |
| CE.MOP= 100372 | CS.STR= 000004  | D\$MAXC 000064  | FS.FNB= 012000   | LF.JRD= 000002  |
| CE.NTE= 100361 | CS.SUP= 000001  | D\$MAXH 000066  | FS.EXI= 001000   | LF.MMF= 000200  |
| CE.RTE= 100376 | CS.TP = 020000  | D\$MAXV 000070  | FS.GET= 006000   | LF.MSF= 000100  |
| CE.SRC= 100364 | CS.V = 000004   | D\$MLL 000040   | FS.HLT= 000000   | LS.DLS= 100000  |
| CE.STP= 100352 | CV\$N_K= 000003 | D\$MXOD 000041  | FS.INI= 000000   | LS.FCC= 000004  |
| CE.TME= 100354 | CV\$31 = 000001 | D\$NA 000062    | FS.KI= 000000    | LS.FCO= 000001  |
| CE.TMO= 100354 | CV\$32 = 000000 | D\$NBEA 000056  | FS.LCL= 100000   | LS.FCI= 000002  |
| CE.UNS= 100354 | CV\$40 = 000002 | D\$NBRA 000054  | FS.LTM= 001000   | LS.ILS= 100000  |
| CF.CHN= 000354 | CS\$ORE= 000400 | D\$NCND= 000054 | FS.MNT= 004000   | LS.MAK= 000020  |
| CF.EOM= 000354 | CS\$RSH= 177564 | D\$NLN 000030   | FS.MSN= 014000   | LS.MNK= 000040  |
| CF.HDR= 000354 | C.ADD 000034    | D\$NN 000060    | FS.REA= 001000   | LS.RES= 000360  |
| CF.LB = 000354 | C.BID 000003    | D\$OUTI 000043  | FS.RET= 000000   | LS.RSV= 000300  |
| CF.LIN= 000354 | C.BUF 000014    | D\$RETF 000050  | FS.REZ= 003000   | LT.CCA= 000020  |

```

243 .SB*TL Add optional data to message
244 **--ADDOPT-Add optional data to message
245 Append the optional data field to a session control message.
246 -
247 Inputs:
248 R2 = Address of next available byte in the message
249 R3 = Virtual address of LLT
250 R4 = Address of CCB
251 R5 = Address of database descriptor
252 Outputs:
253 R2 = Address of next available byte in the message
254 ADDOPT::SAVRG <R3> ; Save address of LLT
255 MAPLLT ; Map to the LLT
256 000426 CLR $BYTE ; Set up optional data byte count
257 000430 MOVB L.OPDL(R3), $BYTE;
258 000436 COUNT$ E$NBS ; Count ECL bytes sent
259 000442 CLRB L.OPDL(R3) ; No optional data any more
260 000446 105063 000102
261 000450 MAP C.BUF(R4) ; Restore mapping to buffer
262 000456 MOVB $BYTE, (R2)+ ; Fill in byte count in image field
263 000462 BEQ 10$; If EQ, none
264 000466 MOV N$LLTM(R5), -(SP); Set up LL1 mapping
265 000470 ADD #L.OPD, R3 ; Point to optional data
266 000474 MOV R3, -(SP) ;
267 000478 MOV $BYTE, R3 ; Set up byte count
268 000482 JSR R1, @MVFBF ; Copy optional data to buffer
269 000486 016546 000024
270 000490 062703 000103
271 000494 010346
272 000498 000510 016703 000000G
273 000502 004177 000000G
274 000506 10$: RESRG <R3> ; Recover LLT address
275 000510 RETURN
276 000514
277 000518

```

```

724 .SBTTL Copy image field
725 +
726 **--CPYIMG-Copy image field
727 Copy an image field with range checking.
728 -
729 Calling sequence:
730 JSR R3,CPYIMG
731 .BYTE <Offset in descriptor>
732 .BYTE <Max field size>
733
734 Inputs:
735 R2 = Address of source image field
736 R4 = Address of pending connect CCB
737
738 Outputs:
739 R2 = Moved past image field
740 'C' Clear - Valid image field processed
741 'C' Set - Image field size invalid
742
743 Registers modified:
744 R0, R1
745
746 CPYIMG: CLR R0 ; Get offset into descriptor
747 BISB (R3)+,R0 ; ...
748 ADD C,BUF+2(R4),R0 ; Compute destination address
749 MOVSB (R3)+,R1 ; Get max length of field
750 CALL PROIMG ; Process the image field
751
752 RTS R3
753
754 001662 005000
755 001664 152300
756 001666 066400 000016
757 001672 112301
758 001674
759 001700 010203

```

```

1193 .SBITL Map object number to task name
1194
1195 ;+
1196 ;*-MAPOBJ-Map object number to task name
1197 Scan the object name table to find the entry for a specified
1198 object type.
1199
1200 Inputs:
1201 R1 = Object number (low byte)
1202
1203 Outputs:
1204 R1 = Object flags and access level
1205 'C' Clear - Object table entry found
1206 'C' Set - Object table entry not found
1207
1208 Registers modified:
1209 R0
1210
1211 002430 MAPOBJ::SAVMAP ; Save current mapping
1212 002434 016700 000000G MOV OBJHD,R0 ; Point to object table listhead
1213
1214 002440 011046 10$: MOV (R0),-(SP) ; Get address of next entry
1215 002442 CALL @CEACC ; Gain access to the block
1216 002446 012600 MOV (SP)+,R0 ; Recover address after mapping
1217 002450 001416 BEQ 30$; If EQ, end of list
1218
1219 002452 120160 000002 CMPB R1,0.TYP(R0) ; Is this the correct entry?
1220 002456 001370 BNE 10$; If NE, no ... keep looking
1221
1222 002460 105701 TSTB R1 ; Is this object 0?
1223 002462 001406 BEQ 20$; If EQ, yes ... we already have the task name etc
1224
1225 .IF DF N$SMCP
1226
1227 MOVB 0.MXC(R0),$R0CPY; Set max # of active copies allowed
1228
1229 .ENDC
1230
1231 002464 016067 070006 000000G MOV 0.NAM(R0),$RQNAM; Fill in task name
1232 002472 016067 000010 000002G MOV 0.NAM+2(R0),$RQNAM+2
1233 002500 116001 000003 20$: MOVB 0.FLG(R0),R1 ; Get flags and access level
1234
1235 002504 005727 30$: TST (PC)+ ; Indicate entry found
1236 002506 000261 SEC ; Indicate error
1237 002510 RESMAP ; Restore mapping
1238 002514 RETURN

```

```

1677 .WORD SLIDSC ; User disconnect complete
1678 .WORD SLIDIS ; Notify user of disconnect or abort
1679 .WORD NOOP ; Wait for disconnect acknowledgement
1680 .WORD SLIEAC ; Error response to connect accept
1681
1682 .ENDC

```

|     |     |     |      |
|-----|-----|-----|------|
| SES | SYM | SYM | RES  |
|     |     |     | RF   |
|     |     |     | RLS  |
|     |     |     | RMV  |
|     |     |     | RSN  |
|     |     |     | RTS  |
|     |     |     | RTS  |
|     |     |     | RSS  |
|     |     |     | RSS  |
|     |     |     | R.A  |
|     |     |     | R.F  |
|     |     |     | R.L  |
|     |     |     | R.L  |
|     |     |     | R.N  |
|     |     |     | SAV  |
|     |     |     | SCS  |
|     |     |     | SCS  |
|     |     |     | SCS  |
|     |     |     | SEN  |
|     |     |     | SPR  |
|     |     |     | ST\$ |
|     |     |     | SV\$ |
|     |     |     | SV\$ |
|     |     |     | TKT  |
|     |     |     | TLA  |
|     |     |     | TRM  |
|     |     |     | TRM  |
|     |     |     | TRM  |
|     |     |     | T\$F |
|     |     |     | T\$L |
|     |     |     | T\$L |
|     |     |     | T\$L |
|     |     |     | T\$L |
|     |     |     | T\$L |
|     |     |     | T\$L |
|     |     |     | T\$L |
|     |     |     | T\$L |
|     |     |     | T\$M |
|     |     |     | T\$M |
|     |     |     | T\$M |
|     |     |     | T\$M |

SES

SESSUB11S CREATED BY MACRO ON 28-JUN-85 AT 20:00

PAGE 8

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES               |
|---------|------------|--------------------------|
| RESVRO  | = 000015   | #6-54                    |
| RF.LOO  | = 100000   | #6-57                    |
| RLSCI   | 003144 RG  | 34-1366 #37-1483 40-1613 |
| RMVWND  | 003176 RG  | #38-1505                 |
| RSNDDI  | = ***** GX | 41-1656                  |
| RT\$INI | = 000007   | #6-54                    |
| RT\$OFF | = 000001   | #6-54                    |
| RT\$ON  | = 000000   | #6-54                    |
| R\$SEIS | = *****    | 14-423 15-481 16-554     |
| R\$SPRO | = *****    | 24-998 25-1016           |
| R.ADD   | 000010     | #6-57 17-649 17-651      |
| R.FLAG  | 000012     | #6-57                    |
| R.LEN   | 000014     | #6-57                    |
| R.LNK   | 000000     | #6-57                    |
| R.NAM   | 000002     | #6-57 17-654             |
| SAVOPT  | 003276 RG  | #39-1551                 |
| SC\$OFF | = 000001   | #6-54                    |
| SC\$ON  | = 000000   | #6-54                    |
| SC\$RST | = 000003   | #6-54                    |
| SC\$SHU | = 000002   | #6-54                    |
| SENDI   | = ***** GX | 41-1655                  |
| SPRST   | = ***** GX | 9-238                    |
| ST\$DIP | = 000012   | 29-1181                  |
| SV\$DUM | = 000001   | #6-54                    |
| SV\$LOA | = 000000   | #6-54                    |
| TKTCB   | = ***** GX | 28-1152                  |
| TLCHK   | 003332 RG  | #40-1583                 |
| TRMLNK  | 003450 RG  | 29-1190 #41-1634         |
| TRMNET  | 003454 RG  | #41-1636                 |
| TRMUSR  | 003532 RG  | 41-1634 #42-1698         |
| T\$FLAG | 000044     | #6-53                    |
| T\$LIF  | 000013     | #6-53                    |
| T\$LIFL | 000013     | #6-53                    |
| T\$LIFO | 000013     | #6-53                    |
| T\$LIFS | 000013     | #6-53                    |
| T\$LIN  | 000000     | #6-53                    |
| T\$LIPS | 000006     | #6-53                    |
| T\$LLD  | 000012     | #6-53                    |
| T\$LLDC | 000045     | #6-53                    |
| T\$LLDL | 000012     | #6-53                    |
| T\$LLDO | 000012     | #6-53                    |
| T\$LLDS | 000012     | #6-53                    |
| T\$LLEN | 000046     | #6-53                    |
| T\$LOPR | 000002     | #6-53                    |
| T\$LTCL | 000024     | #6-53                    |
| T\$LTIM | 000026     | #6-53                    |
| T\$LTPR | 000014     | #6-53                    |
| T\$LTPS | 000020     | #6-53                    |
| T\$NAPL | 000004     | #6-53                    |
| T\$NFE  | 000000     | #6-53                    |
| T\$NLEN | 000010     | #6-53                    |
| T\$NNUL | 000002     | #6-53                    |

SESSUB11S CREATED BY MACRO ON 28-JUN-85 AT 20:00

PAGE 9

```

1 .TITLE SESTIM - Session control timer service
2 .IDENT /V05.00/
3 .ENABL LC

```

```

4 :
5 : Copyright (C) 1982, 1983, 1985 by
6 : Digital Equipment Corporation, Maynard, MASS.

```

```

7 : This software is furnished under a license for use only on a
8 : single computer system and may be copied only with the
9 : inclusion of the above copyright notice. This software, or
10 : any other copies thereof, may not be provided or otherwise
11 : made available to any other person except for use on such
12 : system and to one who agrees to these license terms. Title
13 : to and ownership of the software shall at all times remain
14 : in DEC.

```

```

15 :
16 : The information in this document is subject to change without
17 : notice and should not be construed as a commitment by Digital
18 : Equipment Corporation.

```

```

19 :
20 : DEC assumes no responsibility for the use or reliability of
21 : its software on equipment which is not supplied by DEC.

```

```

22 :
23 : Module description

```

```

24 : Session control timer service

```

```

25 :
26 : Ident history:

```

```

27 :
28 : 4.00 07-NOV-83
29 : DECNET-11M V4.0
30 : DECNET-11M-PLUS V2.0

```

```

31 :
32 : 5.00 22-JUL-85
33 : DECnet-11M/S V4.2
34 : DECnet-11M-Plus V3.0
35 : DECnet-Micro/Rsx V1.0
36 :
37 :
38 :
39 :
40 :

```



```

591 .SBTTL Generate multi-copy task name
592 :+
593 **--MUTNAM-Generate multi-copy task name
594 :
595 Generate a new name for a multi-copy task of the form 'XXX.NN' where:
596 :
597 XXX is the first three characters of the prototype's name
598 NN is an octal number in the range 0 to 77
599 :-
600 Inputs:
601 R1 = Copy number for generating task name
602 2(SP) - First word of task name
603 4(SP) - Second word of task name
604 :
605 Registers modified:
606 R0
607 :
608 MUTNAM: SAVRG <R1>
609 MOV #*R1, 6(SP) ; Initialise second word of task name
610 MOV R1,R0 ; Copy number
611 BIC #*C<7>,R1 ; Isolate low octal digit
612 ADD #36,R1 ; Add in RAD50 '0' offset
613 ASR R0 ; Get high octal digit
614 ASR R0 ; ...
615 ASR R0 ; ...
616 BEQ 10$; If EQ, no high digit
617 :
618 ADD R1,6(SP) ; Add low digit to name
619 MOV R0,R1 ; Get high digit
620 ADD #36,R1 ; Add in RAD50 '0' offset
621 :
622 10$: .IF DF R0$EIS
623 :
624 MUL #50,R1 ; Convert to second digit value
625 :
626 .IFF
627 :
628 MOV #50,R0 ; Set up multiplier value
629 CALL @CEMUL ; Convert to second digit value
630 :
631 .ENDC
632 :
633 ADD R1,6(SP) ; Finish off task name
634 RESRG <R1> ; Recover copy number
635 RETURN
636 :
637 .ENDC

```

```

115 000150 010560 000000G MOV R5,U.VCB(R0) ; Set up address of dummy
116 000154 062760 000000G ADD #N$VCB,U.VCB(R0); volume control block
117 000162 017760 000000G MOV @TKTCB,U.ACP(R0); Set up ACP's TCB address
118 000170 142760 000000C BICB #US.MNT:US.MDM,U.STS(R0)
119 000176 142760 000000G BICR #US.OFL,U.ST2(R0)
120
121 000204 152765 000040 000005 BISB #NFSMOU,NFLG(R5)
122
123 000212 016701 000000G 20$: MOV TINS,R1 ; Randomise logical link address encoding
124 000216 016100 177776 MOV -2(R1),R0 ; (by adding seconds and ticks)
125 000222 061100 ADD (R1),R0
126 000224 150067 000001G BISB R0,$ENCOD+1
127 000230 017700 000000G MOV @DECPTR,R0 ; Get the segment size
128 000234 016001 000036 MOV D$SEG(R0),R1
129 000240 001005 BNE 30$
130 000242 017701 000000G MOV @RDBSZ,R1 ; If NE - its already set in (ETAB
131 000246 162701 000022 SUB #N$SOVR,R1 ; Else, compute our segment size
132 000252 000403 BR 40$; and base it on RDB size
133 000254 020001 000036 30$: CMP D$SEG(R0),R1 ; Is it too large ?
134 000260 002402 BLT 50$; If LT, its ok
135
136 000262 010160 000036 40$: MOV R1,D$SEG(R0) ; Else, reset it to largest possible value
137 ; Store the segment size
138 000266 005727 50$: IST (PC)+ ; Indicate success
139 000270 000261 100$: SEC ; Indicate failure
140 000272 RETURN

```

\*\*FILE\*\*ID\*\*SESMN

```

SSSSSSSS EEEEEEEEE SSSSSSSS MM MM NN NN
SSSSSSSS EEEEEEEEE SSSSSSSS MM MM NN NN
SS EEEEEEEEE SS SSSSSSSS MMMM MMMM NN NN
SS EEEEEEEEE SS SSSSSSSS MMMM MMMM NN NN
SS EEEEEEEEE SS SSSSSSSS MM MM NN NN
SS EEEEEEEEE SS SSSSSSSS MM MM NN NN
SSSSSSSS EEEEEEEEE SSSSSSSS MM MM NN NN
SSSSSSSS EEEEEEEEE SSSSSSSS MM MM NN NN
SS EEEEEEEEE SS SSSSSSSS MM MM NN NN
SS EEEEEEEEE SS SSSSSSSS MM MM NN NN
SSSSSSSS EEEEEEEEE SSSSSSSS MM MM NN NN
SSSSSSSS EEEEEEEEE SSSSSSSS MM MM NN NN

```

```

....
....
....
....

```

```

11 11 SSSSSSSS
11 11 SSSSSSSS
1111 1111 SS
1111 1111 SS
11 11 SS
11 11 SSSSSS
11 11 SSSSSS
11 1 SS
11 1 SS
11 11 SS
11 11 SS
111111 111111 SSSSSSSS
111111 111111 SSSSSSSS

```

| SYMBOL  | VALUE    | REFERENCES |
|---------|----------|------------|
| E\$MOD  | 000012   | #6-54      |
| E\$NOD  | 000010   | #6-54      |
| E\$PORT | 000014   | #6-54      |
| E\$PRM  | 000002   | #6-54      |
| E\$STAT | 000006   | #6-54      |
| E\$TCB  | 000004   | #6-54      |
| E.CTL   | 000020   | #6-54      |
| E.DATA  | 000046   | #6-54      |
| E.EVT   | 000002   | #6-54      |
| E.LCN   | 000042   | #6-54      |
| E.LEN   | 000216   | #6-54      |
| E.LIN   | 000024   | #6-54      |
| E.LNK   | 000000   | #6-54      |
| E.MOD   | 000036   | #6-54      |
| E.NOD   | 000034   | #6-54      |
| E.PDV   | 000021   | #6-54      |
| E.PORT  | 000040   | #6-54      |
| E.PRM   | 000026   | #6-54      |
| E.PVC   | 000044   | #6-54      |
| E.SIZ   | 000022   | #6-54      |
| E.TIME  | 000004   | #6-54      |
| FR\$BCC | = 000007 | #6-54      |
| FR\$CCF | = 000001 | #6-54      |
| FR\$CDF | = 000002 | #6-54      |
| FR\$DAO | = 000011 | #6-54      |
| FR\$EXC | = 000000 | #6-54      |
| FR\$FRM | = 000010 | #6-54      |
| FR\$FTL | = 000005 | #6-54      |
| FR\$OPN | = 000004 | #6-54      |
| FR\$RFD | = 000006 | #6-54      |
| FR\$SBU | = 000012 | #6-54      |
| FR\$SHQ | = 000003 | #6-54      |
| FR\$UBU | = 000013 | #6-54      |
| FR\$UPT | = 000014 | #6-54      |
| IE.UPN  | = *****  | 9-144      |
| MO\$SAC | = 000016 | #6-54      |
| MO\$SPR | = 000012 | #6-54      |
| MO\$SSV | = 000014 | #6-54      |
| MO\$25A | = 000006 | #6-54      |
| MO\$25P | = 000002 | #6-54      |
| MO\$25S | = 000004 | #6-54      |
| MO\$29S | = 000010 | #6-54      |
| M\$HIGH | = 000003 | #6-54      |
| M\$3100 | = 000000 | #6-54      |
| M\$3101 | = 000001 | #6-54      |
| M\$3102 | = 000002 | #6-54      |
| M\$3103 | = 000003 | #6-54      |
| NF\$DMO | = 000010 | 8-94       |
| NF\$MQU | = 000040 | 8-122      |
| NF\$SHU | = 000004 | 8-116      |
| NF\$TIM | = 000200 | 8-105      |
| N\$ACTL | 000032   | 8-97       |

GX

6-54  
8-122  
8-122

E 5

```

275 .SBTTL Process received CC message
276
277 + **--RCVCC-Process received CC message
278
279 This routine processes received CC messages.
280
281 -
282 Inputs:
283 R2 = Pointer to field following source link address
284 R4 = Address of CB
285 R5 = Address of database descriptor
286
287 RCVCC: MOVB (R2)+,$SRVCS ; Get services requested
288 MOVB (R2)+,$INFO ; info field (NSP version #)
289 10$: TSTB (R2)+ ; ...
290 BMI 10$; ...
291 MOVB (R2)+,$SEGMT ; segment size
292 MOVB (R2)+,$SEGMT+1 ; ...
293
294 MOVB $SRVCS,R1 ; Get requested services
295 BIT #*C<CL$SFL!CL$MFL!CL$TYP>,R1
296 BNE 100$; If NE, invalid CC message
297 CMPB #CL$SFL!CL$MFL!CL$TYP,R1
298 BEQ 100$; Can't ask for message and segment flow control together
299
300 CALL GETOPT ; Get optional data from message
301 BCS 100$; If CS, format error
302
303 CALLE FNDLLT ; Try to find LLT for this link
304 BCS 110$; If CS, none present
305
306 CMPB #ST$CIS,(R3) ; Are we waiting for a CC on this link?
307 BNE 90$; If NE, no
308
309 .IF DF N$$NCT
310 MOV $OPLNG,$BYTE ; Count bytes received for CC message
311 COUNT$ E$NBR ; ...
312 COUNT$ E$NMR ; Count message received
313 .ENDC
314
315 MOV N$$SLA(R5),L,RLA(R3)
316 CLR L,TMRD(R3) ; Stop CI timer
317 CALLE FNDDLY ; Find initial estimate of delay to remote node
318
319 CALL PROCON ; Process requested connect services
320 CALLE STOPCI ; Stop CI retransmission
321
322 .IF DF N$$SLI
323
324 BIT #LT,SLI*400,(R3) ; Is this an SLI link?
325 BEQ 20$; If EQ, no
326 CALL SLICC ; Tell system level interface link has been confirmed
327 BR 30$; Return to common code
328
329 .ENDC
330
331 20$: CALL USRCC ; Tell user link has been confirmed

```

F 5

| SYMBOL  | VALUE        | REFERENCES                                            |
|---------|--------------|-------------------------------------------------------|
| ACCLLT  | = ***** GX   | 10-264                                                |
| ACKCI   | = ***** GX   | 9-216                                                 |
| ADDLNK  | = ***** GX   | 9-188                                                 |
| ADDOPT  | = ***** GX   | 16-567                                                |
| BRKLNK  | = ***** GX   | 10-271                                                |
| CEACC   | = ***** GX   | 9-177                                                 |
| CL\$MFL | = 000010     | 9-165 9-167 11-294 11-296 15-525                      |
| CL\$SFL | = 000004     | 9-165 9-167 11-294 11-296 15-521                      |
| CL\$TYP | = 000001     | 9-165 9-167 11-294 11-296                             |
| CNVCI   | = ***** GX   | 9-201                                                 |
| CTRSES  | = ***** GX   | 9-197 9-198 11-310 11-311 12-380 12-381 12-382 13-422 |
| CV\$MSK | = 000003     | 15-513                                                |
| CV\$40  | = 000002     | 15-514                                                |
| C.ADD   | 000034       | 15-509                                                |
| C.BUF   | 000014       | 8-101 8-102 9-186 18-616 *18-618 18-621 18-622        |
| C.BUF2  | 000024       | 18-618                                                |
| C.CNT   | 000020       | 18-617                                                |
| C.FLG2  | 000032       | 8-100                                                 |
| C.LI.W  | 000006       | *9-205 15-505                                         |
| C.MOD   | 000011       | 18-619                                                |
| C.STS   | 000012       | 9-149 9-170                                           |
| DCSTA   | 001304 R     | 13-433 #13-443                                        |
| DECPT   | = ***** R GX | 11-334 15-500 15-529 16-572                           |
| DISTA   | 001162 R     | 12-385 #12-398                                        |
| D\$INAC | 000044       | 11-335                                                |
| D\$LNUM | 000014       | 15-501                                                |
| D\$RETF | 000050       | 16-573                                                |
| D\$SEG  | 000036       | 15-530 15-532                                         |
| ER\$ABO | = 000046     | 13-428                                                |
| ER\$CDI | = 000052     | 12-374                                                |
| ER\$COM | = 000047     | 10-270                                                |
| ER\$RES | = 000001     | 9-224                                                 |
| ER\$STA | = 000051     | 12-426                                                |
| E\$NBR  | 000014       | #6-51 11-310 12-380                                   |
| E\$NBS  | 000020       | #6-51                                                 |
| E\$NCR  | 000034       | #6-51 9-198                                           |
| E\$NCS  | 000036       | #6-51                                                 |
| E\$NIC  | 000044       | #6-51                                                 |
| E\$NLEN | 000050       | #6-51                                                 |
| E\$NLLA | 000012       | #6-51                                                 |
| E\$NLNK | 000000       | #6-51                                                 |
| E\$NML  | 000040       | #6-51                                                 |
| E\$NMR  | 000024       | #6-51 9-197 11-311 12-381 13-422                      |
| E\$NMS  | 000030       | #6-51 12-382                                          |
| E\$NNOD | 000002       | #6-51                                                 |
| E\$NRT  | 000042       | #6-51                                                 |
| E\$NRTF | 000005       | #6-51                                                 |
| E\$NSEG | 000010       | #6-51                                                 |
| E\$NTIM | 000046       | #6-51                                                 |
| E\$NUSE | 000004       | #6-51                                                 |
| E\$STRT | 000006       | #6-51                                                 |
| FLSRES  | = ***** GX   | 8-118                                                 |

```

194 .SBTTL User disconnect completion
195
196 ;+
197 ;**--USRDSC--User disconnect completion
198 ; Complete the user disconnect request.
199 ;
200 ; Inputs:
201 ; R3 = Virtual address of LLT
202 ; R5 = Address of database descriptor
203 ;
204 ; Registers modified:
205 ; R0, R1, R2, R4
206
207 000312 122763 000000 000037 USRDSC::CMPB #NS$DON,L.CSTA(R3)
208 000320 001020 ; BNE 10$; Wait till network processing complete
209
210 000322 016300 000040 MOV L.WIND(R3),R0 ; Get window block address
211 000326 105760 000014 TSTB W.KAST(R0) ; Are there any outstanding ASTs?
212 000332 001013 ; BNE 10$; BR if yes
213 000334 112763 000000 000036 MOVB #US$DON,L.USTA(R3)
214 000342 016746 000000G MOV $IOPKT,-(SP) ; Save any I/O packet
215 000346 CALL RMVLNK ; Remove link resources
216 000352 CALL IOSUC ; Complete the disconnect request
217 000356 012667 000000G MOV (SP)+,$IOPKT ; Restore any I/O packet
218 000362 10$: RETURN

```

SESSL1 - Session control system MACRO V05.03b Friday 28-Jun-85 19:58 Page 6  
Macro definitions

```

42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,MAPLLT,CALLE
45 .MCALL SLIDF$,CCBDF$,ECDDB$,PDVDF$,MSGDF$,CNBDF$,RNBDF$
46 .MCALL DHBDF$
47
48 000000 SLIDF$; Define SLI symbols
49 000000 CCBDF$; Define CCB offsets
50 000000 ECDDB$; Define ECL database offsets
51 000000 PDVDF$; Define PDV offsets
52 000000 MSGDF$; Define message symbols
53 000000 CNBDF$; Define connect request block offsets
54 000000 RNBDF$; Define remote name block offsets
55 000000 DHBDF$; Define the DEC home block offsets
56
57 000001 N$$SES = 1 ; This module is part of session control

```

SESSL1 - Session control system MACRO V05.03b Friday 28-Jun-85 19:58 Page 7



```

519 MOVB (R2)+,(R0)+ ;
520 MOVB (R2)+,R3 ; Get length of node name
521 MOVB R3,(R0)+ ; and plant in output buffer
522 BEQ 20$; If EQ, null node name
523
524 10$: MOVB (R2),(R0)+ ; Copy node name to output buffer
525 MOVB (R2)+,(R1)+ ; and to local copy
526 SOB R3,10$;
527
528 20$: TST $WORK ; Did caller specify a node address?
529 BEQ 30$; If EQ, no
530 CALL FNDNAM ; Find mapping by name
531 BR 40$; and exit via common path
532
533 30$: CALL FNDADD ; Find mapping by node address
534 40$: BCS 50$; If CS, no mapping found
535 MOVB $WORK+2,R0 ; Get length of node name
536 ADD #$WORK+2,R0 ; Point past end of node name
537 CLC ; Indicate mapping found
538 50$: RESRG <R1,R2> ; Restore registers
539 RETURN
540
541 ;+
542 ; **--GLNHOS-Get host node address and name
543 ;--
544 .PSECT $HIGH
545
546 GLNHOS: MOV D$HOST(R1),(R0)+ ; Fill in host node address
547 MOV #HOST,R3 ; Point to host logical node name
548 CALL NODIMG ; Convert to image field
549 CLC ; Indicate success
550 RETURN

```

|         |        |          |        |          |        |          |        |          |        |
|---------|--------|----------|--------|----------|--------|----------|--------|----------|--------|
| LT.DIR= | 000010 | MA.DA =  | 000000 | N\$DLY   | 000014 | N.SNM    | 000046 | S.EERR=  | 100210 |
| LT.LCL= | 000001 | MA.LL =  | 000020 | N\$ELEN  | 000054 | N.SNMC   | 000044 | S.EIDM=  | 100214 |
| LT.LPL= | 000002 | MC.CC =  | 000040 | N\$ENC   | 000042 | N.SOT    | 000037 | S.EINF=  | 100212 |
| LT.NOT= | 000040 | MC.CI =  | 000020 | N\$ERRC  | 000022 | N.SUS    | 000042 | S.EIOF=  | 100373 |
| LT.RSU= | 000200 | MC.DC =  | 000100 | N\$FLG   | 000005 | P\$P45=  | 000000 | S.ELNS=  | 100365 |
| LT.SLI= | 000004 | MC.DI =  | 000060 | N\$FNC   | 000006 | P\$SWRD= | 000000 | S.ELST=  | 100216 |
| LT.TDA= | 000100 | MC.NO =  | 000000 | N\$GENQ  | 000052 | Q\$OPT=  | 000010 | S.ELWS=  | 100220 |
| L\$ASG= | 000000 | MC.RC =  | 000140 | N\$GTM   | 000015 | RF.LOO=  | 100000 | S.EMTL=  | 100222 |
| L\$DRV= | 000000 | MD.BM =  | 000040 | N\$HIGH  | 000033 | R\$DER=  | 000000 | S.ENOF=  | 100224 |
| L\$P11= | 000001 | MD.EM =  | 000100 | N\$LLT   | 000026 | R\$K11=  | 000001 | S.ENRO=  | 100332 |
| L\$11R= | 000000 | MD.ILS=  | 000040 | N\$LLTM  | 000024 | R\$SND=  | 000000 | S.ENSL=  | 100327 |
| L.CSTA  | 000037 | MD.IM =  | 000020 | N\$LVC   | 000036 | R\$11M=  | 000000 | S.ENUR=  | 100331 |
| L.CTR   | 000074 | MF.ACK=  | 000004 | N\$MBXQ  | 000050 | R\$11S=  | 000000 | S.EOTB=  | 100372 |
| L.DCR   | 000100 | MF.CTL=  | 000010 | N\$PLLT  | 000030 | R.ADD    | 000010 | S.ERBO=  | 100226 |
| L.FLAG  | 000014 | MF.DAT=  | 000000 | N\$SLA   | 000016 | R.FLAG   | 000012 | S.ERES=  | 100377 |
| L.ILS   | 000052 | M\$CRB=  | 000124 | N\$SNOD  | 000012 | R.LEN    | 000014 | S.ERNIS= | 100375 |
| L.LTT   | 000066 | M\$CRX=  | 000000 | N\$TIM   | 000004 | R.LNK    | 000000 | S.ETMI=  | 100230 |
| L.LDA   | 000032 | M\$FCS=  | 000000 | N\$VCB   | 000010 | R.NAM    | 000002 | S.EUNN=  | 100376 |
| L.LIA   | 000034 | M\$MGE=  | 000000 | N\$SACC= | 000001 | STCC =   | 000004 | S.EURO=  | 100374 |
| L.LLA   | 000002 | M\$MUP=  | 000000 | N\$SACK= | 000011 | STCIR=   | 000006 | S.SEOM=  | 000003 |
| L.LNG   | 000124 | M\$NET=  | 000000 | N\$SEVL= | 000001 | STCIS=   | 000002 | S.SSUC=  | 000001 |
| L.LNO   | 000026 | M\$OVR=  | 000000 | N\$SHDR= | 000007 | STDAT=   | 000010 | T\$KMG=  | 000000 |
| L.LPT   | 000065 | NC.FM0=  | 000000 | N\$SLDV= | 000001 | STDIP=   | 000012 | T\$MIN=  | 000000 |
| L.LSA   | 000030 | NC.FM1=  | 000001 | N\$SMLL= | 000001 | STPND=   | 000014 | US\$CNF= | 000002 |
| L.LSFD  | 000046 | NC.FM2=  | 000002 | N\$SMOV= | 000010 | SABO =   | 000022 | US\$DIS= | 000006 |
| L.LSFI  | 000044 | NF\$BLK= | 000100 | N\$SNCT= | 000001 | SACC =   | 000002 | US\$DON= | 000000 |
| L.LTT   | 000062 | NF\$DMO= | 000010 | N\$SOVR= | 000022 | SBUF =   | 000026 | US\$DSC= | 000004 |
| L.MASQ  | 000070 | NF\$MOU= | 000040 | N\$SPEM= | 000001 | S\$CNR = | 000000 | US\$EAC= | 000012 |
| L.MAST  | 000073 | NF\$RST= | 000002 | N\$SSES= | 000001 | S\$CON = | 000000 | US\$WDS= | 000010 |
| L.MASZ  | 000077 | NF\$SCN= | 000020 | N.CAC    | 000120 | S\$DAT = | 000002 | V\$SCTR= | 001000 |
| L.NIN   | 000020 | NF\$SHU= | 000004 | N.CACC   | 000116 | S\$DIS = | 000020 | X\$SDBT= | 000000 |
| L.NXN   | 000016 | NF\$TIM= | 000200 | N.CBL =  | 000142 | S\$DRQ = | 000012 | ZF.COU=  | 001000 |
| L.NXTH  | 000010 | NM\$ARA= | 176000 | N.CDA    | 000142 | S\$DSR = | 000006 | ZF.DDM=  | 000001 |
| L.OPD   | 000103 | NM\$NOD= | 001777 | N.CDAC   | 000140 | S\$GLN = | 000024 | ZF.DIA=  | 004000 |
| L.OPDL  | 000102 | N\$SDON= | 000000 | N.CDDS   | 000070 | S\$INT = | 000004 | ZF.DLC=  | 000002 |
| L.REM   | 000006 | N\$SDI=  | 000002 | N.CDEV   | 000062 | S\$IRQ = | 000016 | ZF.DVP=  | 100000 |
| L.RFC   | 000050 | N\$WDC=  | 000004 | N.CID    | 000064 | S\$MRQ = | 000010 | ZF.INI=  | 040000 |
| L.RLA   | 000004 | NT\$AKD= | 000020 | N.CIDC   | 000062 | S\$NOT = | 000010 | ZF.KMX=  | 000020 |
| L.RNO   | 000022 | NT\$AKI= | 000022 | N.CPS    | 000106 | S\$NTIF= | 000001 | ZF.LLC=  | 000004 |
| L.RTO   | 000060 | NT\$CC = | 000016 | N.CPSC   | 000104 | S\$PBOM= | 000040 | ZF.LMC=  | 000100 |
| L.RTYD  | 000055 | NT\$CON= | 000000 | N.CTL    | 000000 | S\$PEOM= | 000100 | ZF.MAN=  | 020000 |
| L.RTYI  | 000057 | NT\$CTL= | 000000 | N.CUIC   | 000066 | S\$PMOM= | 000000 | ZF.MFL=  | 000010 |
| L.SEC   | 000064 | NT\$DAT= | 000002 | N.CUNI   | 000064 | S\$PSMG= | 000200 | ZF.MTM=  | 000400 |
| L.SEGZ  | 000076 | NT\$DC = | 000012 | N.DDE    | 000010 | S\$PSEG= | 000100 | ZF.MUX=  | 000040 |
| L.STA   | 000000 | NT\$DIS= | 000014 | N.DDEC   | 000006 | S\$REJ = | 000004 | ZF.PSE=  | 002000 |
| L.TC    | 000042 | NT\$DLS= | 000006 | N.DFM    | 000004 | S\$SND = | 000006 | ZF.SLI=  | 010000 |
| L.TIC   | 000043 | NT\$ILS= | 000010 | N.DGP    | 000006 | S\$SNI = | 000014 | ZF.TIM=  | 000200 |
| L.TIPD  | 000013 | NT\$IMS= | 000002 | N.DNM    | 000014 | S\$TMDA= | 000002 | ZF.XSP=  | 000000 |
| L.TIPI  | 000012 | NT\$INT= | 000004 | N.DNMC   | 000012 | S\$WGR=  | 000000 | ZS.ASN=  | 100000 |
| L.TMRD  | 000054 | NT\$RET= | 000032 | N.DOT    | 000005 | S\$YSZ=  | 007600 | ZS.BSY=  | 140000 |
| L.TMRI  | 000056 | NT\$RGU= | 000024 | N.DUS    | 000010 | S.EABL=  | 100200 | Z.AVL    | 000014 |
| L.TYP   | 000001 | NT\$RTR= | 000030 | N.SDE    | 000042 | S.EABM=  | 100370 | Z.DAT    | 000016 |
| L.USA   | 000224 | NT\$TSP= | 000026 | N.SDEC   | 000040 | S.EABO=  | 100367 | Z.DSP    | 000000 |
| L.USTA  | 000036 | NSACQ    | 000000 | N.SEGZ   | 000002 | S.EABS=  | 100202 | Z.FLG    | 000010 |
| L.VER   | 000015 | NSACTL   | 000032 | N.SFM    | 000036 | S.EACR=  | 100336 | Z.LEN =  | 000016 |
| L.WIND  | 000040 | N\$CIR   | 000034 | N.SGP    | 000040 | S.ECBE=  | 100204 | Z.LLN    | 000006 |
| MA.CI = | 000040 | N\$DLA   | 000020 | N.SND    | 000030 | S.EDBO=  | 100206 | Z.MAP    | 000020 |

```

279 .SBTTL Allocate an ECL node database
280 +
281 **--ALLDB--Allocate an ECL node database
282
283 Allocate an ECL node database for the remote node. If an entry is
284 found for this node, use it.
285
286 Inputs:
287 R3 = Virtual address of LLT
288 R5 = Address of database descriptor
289
290 Outputs:
291 'C' Clear - ECL node database allocated
292 'C' Set - Failed to allocate database
293
294 Registers modified:
295 R0, R1
296
297 ALLDB:: SAVRG <R4> ; Get a free register
298 MAP N$ENC+4(R5) ; Map to the ECL database area
299 MOV R5,R4 ; Compute address of database listhead
300 ADD #N$ENC,R4 ; ...
301
302 CLR R0 ; Initialise new block pointer
303 MOV (R4),R4 ; Get address of next database
304 BEQ 20$; If EQ, no more
305 CMP N$SNOD(R5),E$NNOD(R4)
306 BEQ 50$; If EQ, use this one
307 TSTB E$NUSE(R4) ; Is this database in use?
308 BNE 10$; If NE, yes ... keep looking
309 TST R0 ; Do we already have a free block?
310 BNE 10$; If NE, yes
311 MOV R4,R0 ; Save this entry
312 BR 10$
313
314 MOV R0,R4 ; Copy address of database
315 BEQ 100$; If EQ, none available
316
317 .IF DF N$EVL
318
319 MOV EVDSC,-(SP)
320 ADD #E$NOD,(SP)
321 MOV E$NNOD(R0),a(SP)+
322 BEQ 30$; If EQ, database was not previously in use
323
324 .ENDC
325
326 EVT$ 3,2,<NOD>,E$NLEN; Log database reused
327
328 TST (R0)+ ; Skip link word
329 MOV #<E$NLEN-2>/2,R1; Clear out the database
330 CLR (R0)+ ; ...
331 SOB R1,40$; ...
332
333 MOV @ZTIM2,E$NTIM(R4) ; Get time since midnight
334 MOV @DECPT,R1 ; Get DEC home block address
335 MOV D$OUTT(R1),E$NRTPT(R4)

```

```

756 .SBTTL Copy optional outgoing data
757
758 *--CPYOPT-Copy optional outgoing data
759
760 Copy optional outgoing data to the local optional data buffer.
761
762 Calling sequence:
763 JSR R1,CPYOPT
764 .WORD <Offset of buffer descriptor in I/O packet>
765
766 Inputs:
767 R3 = Virtual address of LLT
768 R5 = Address of database descriptor
769
770 Registers modified:
771 R0, R2
772
773 001702 CPYOPT::SAVRG <R3> ; Save address of LLT
774 001704 012702 000000G MOV #DPLNG,R2 ; Point to internal optional data buffer
775
776 001710 016700 000000G MOV $IOPKT,R0 ; Compute address of buffer
777 001714 062100 ADD (R1),R0 ; descriptor in I/O packet
778 001716 016003 000004 MOV 4(R0),R3 ; Get # of bytes to copy
779 001722 010322 MOV R3,(R2)+ ; Fill in descriptor
780 001724 001406 BEQ 20$; If EQ, none
781 001726 MAP (R0)+ ; Map the user's buffer
782 001732 012000 MOV (R0)+,R0 ; Get virtual address of user's buffer
783 001734 112022 10$: MOVB (R0)+(R2)+ ; Copy the optional data
784 001736 SOB R3,10$; ...
785
786 001742 20$: RESRG <R3> ; Recover LLT address
787 001744 MAPLLT ; and restore it's mapping
788 001752 000201 RTS R1

```

E 13

```

1240 .SBITL Point CCB buffer descriptor at optional data
1241
1242 : **PNTCCB-Point CCB buffer descriptor at optional data
1243 :
1244 : Point the buffer descriptor in the CCB at the optional data in the
1245 : LLT.
1246 :
1247 : Inputs:
1248 : R3 = Virtual address of LLT
1249 : R4 = Address of CCB
1250 : R5 = Address of database descriptor
1251 :
1252 : Registers modified:
1253 : R0
1254 :
1255 002516 PNTCCB::IF DF N$MMLL
1256
1257 002516 016564 000024 000014 MOV N$LLTM(R5),C.BUF(R4)
1258
1259 .IFF
1260
1261 CLR C.BUF(R4) ; LLT is allocated from DSR
1262
1263 .ENDC
1264
1265 002524 010364 000016 MOV R3,C.BUF+2(R4) ; Set up buffer bias and virtual address
1266 002530 062764 000103 000016 ADD #L.OPD,C.BUF+2(R4)
1267 002536 116300 000102 MOV L.OPDL(R3),R0 ; Get length of optional data
1268 002542 010064 000020 MOV R0,C.CNT(R4) ; to CCB count field
1269 002546 105063 000102 CLRB L.OPDL(R3) ; No optional data now in LLT
1270 002552 RETURN

```

F 13

```

1684 .SBTTL Process user disconnect state
1685 ;+
1686 ;**--TRMUSR-Process user disconnect state
1687 ;
1688 ; Terminate the logical link with respect to the user by calling the
1689 ; user state action routine.
1690 ;
1691 ;Inputs:
1692 ; R3 = Virtual address of LLT
1693 ; R5 = Address of database descriptor
1694 ;
1695 ;Registers modified:
1696 ; R0, R1, R4
1697 ;
1698 TRMUSR::SAVRG <R2,R3> ; Save some registers
1699 MOVW L,USTA(R3),R0 ; Get disconnect state for user
1700 ;
1701 .IF DF N$$SLI
1702 ;
1703 BIT "I.SLI*400,(R3)
1704 BEQ 10$; If EQ, not SLI link
1705 ADD #SSTATE-USTATE,R0
1706 10$:
1707 .ENDC
1708 ;
1709 CALL @USTATE(R0) ; Process user disconnect state
1710 RESRG <R3,R2> ; Restore registers
1711 MAPLLT
1712 RETURN
1713 ;
1714 .END

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE       | REFERENCES                                           |
|---------|-------------|------------------------------------------------------|
| T\$NOPL | 000006      | #6-53                                                |
| T\$NRNI | 000042      | #6-53                                                |
| T\$NRPL | 000005      | #6-53                                                |
| T\$NRUL | 000007      | #6-53                                                |
| T\$NVR  | 000001      | #6-53                                                |
| T\$RPRI | 000040      | #6-53                                                |
| T\$SYC  | 000034      | #6-53                                                |
| T\$T5   | 000030      | #6-53                                                |
| T\$T6   | 000032      | #6-53                                                |
| T.PCB   | = ***** GX  | 38-1508                                              |
| T.RCVL  | = ***** GX  | 28-1154 *28-1155                                     |
| UISAR6  | = ***** GX  | 17-660 21-851 23-951                                 |
| USRCNF  | = ***** GX  | 41-1663                                              |
| USRDIS  | = ***** GX  | 41-1665 41-1667                                      |
| USRDSG  | = ***** GX  | 41-1664                                              |
| USTATE  | = 003516 RG | #41-1662 42-1709                                     |
| US\$CNF | = 000002    | 12-374                                               |
| US\$DIS | = 000006    | 12-375 12-377                                        |
| US\$DON | = 000000    | 12-376 41-1642                                       |
| US\$DSC | = 000004    | 12-379                                               |
| WORD1   | = 000302    | #11-326 11-326                                       |
| WORD2   | = 000010    | #11-326 11-326 #11-326 11-326                        |
| W.CINT  | = 000022    | 21-870 *21-872                                       |
| W.LLT   | = 000004    | 38-1526                                              |
| W.LUN   | = 000003    | 38-1520                                              |
| W.MBOX  | = 000012    | 38-1506                                              |
| W.RCVG  | = 000024    | 21-864 *21-866                                       |
| W.SNDQ  | = 000016    | 21-858 *21-860                                       |
| W.WBL   | = 000026    | 38-1532                                              |
| X\$SHDR | = *****     | 38-1511                                              |
| ZTIM2   | = ***** GX  | 11-333                                               |
| \$ALOCX | = ***** GX  | 8-131                                                |
| \$BYTE  | = ***** GX  | *10-261 *10-262 10-267 10-273 *17-702 *17-703 17-704 |
| \$CALLX | = ***** GX  | 8-131 8-198 9-238 34-1365 35-1404 35-1425 40-1602    |
| \$DEACX | = ***** GX  | 8-198 35-1425                                        |
| \$ENCOD | = ***** GX  | 8-173 *8-174                                         |
| \$IOPKT | = ***** GX  | 19-775 40-1586                                       |
| \$LADDR | = ***** GX  | *17-611 *17-618 17-651                               |
| \$MAIBX | = ***** GX  | 9-225 *23-930 24-993 *26-1072 40-1584                |
| \$MENU  | = ***** GX  | *17-678 17-680 17-699                                |
| \$OPDAT | = ***** GX  | 39-1556                                              |
| \$OPLNG | = ***** GX  | 19-773 39-1551                                       |
| \$RCCB  | = ***** GX  | 17-663                                               |
| \$RQNAM | = ***** GX  | *30-1231 *30-1232                                    |
| \$SEGMT | = ***** GX  | 17-666                                               |
| \$SESDB | = ***** GX  | 17-705                                               |
| .CCBRT  | 000334 R    | #23-962 23-970 23-971 23-972                         |

SESTIM - Session control timer MACRO V05.03b Friday 28-Jun-85 20:01 Page 6  
Macro definitions

```

42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,CALLE,MAPLLT
45 .MCALL ECDDB$,LLTDF$,DHBDf$
46
47 DHBDf$; Define DEC home block offsets
48 ECDDB$; Define ECL database offsets
49 LLTDF$; Define LLT offsets
50
51 000001 N$$SES = 1 ; This module is part of session control

```

SESTIM - Session control timer MACRO /05.03b Friday 28-Jun-85 20:01 Page 7



```

639 .SBTTL Complete I/O request
640
641 ;+
642 ;*-IOERR-Complete I/O request in error
643 ;*-IOSUC-Complete I/O request successfully
644 ;*-IODUN1-Complete the I/O request with zero second I/O status word
645 ;*-IODUN-Complete I/O request
646
647 Call the executive to complete an I/O request.
648
649 Inputs:
650 R0 = Contents of first I/O status word (IODUN and IODUN1 only)
651 R1 = Contents of second I/O status word (IODUN only)
652
653 Outputs:
654 'C' Always set
655
656 Registers modified:
657 R2
658
659 .PSECT
660 000454 IOERR:: MOV a(SP)+,R0 ; Get the error code
661 BR IODUN1 ; Enter common code
662
663 000460 IOSUC:: MOV #IS.SUC&377,R0 ; Set successful completion
664
665 000464 IODUN1::CLR R1 ; Second I/O status word = 0
666
667 000466 IODUN:: MOV $IOPKT,R3 ; Recover I/O packet address
668 000472 BEQ 10$; If E0, none
669
670 .IF DF X$$HDR
671
672 MOV I.TCB(R3),R2 ; Get TCB address
673 MOV I.PCB(R2),R2 ; then PCB address
674 MAP P.REL(R2) ; Map the task header
675
676 .ENDC
677
678 000474 BIC #1,a(I.L)2(R3) ; Remove LUN interlock
679 000502 CLR I.PRM+16(R3) ; Clear RMS record locking flag
680 000506 CALL @IOFIN ; Complete the request
681
682 .IF DF X$$HDR
683
684 RECMAP ; Recover APR6 mapping
685
686 .ENDC
687
688 000512 SEC 10$: SEC ; Set C-bit
689 000514 RETURN
690
691 000001 .END

```

```

142 .SBTTL Allocate session control databases
143
144 + **--ALOCDB--Allocate session control databases
145
146 Allocate and initialise the databases required by session control.
147
148 - Outputs:
149 R5 = Address of database descriptor
150 'C' Clear - Databases allocated
151 'C' Set - Failed to allocate databases
152
153 Registers modified:
154 R0, R1, R2, R3, R4, R5
155
156
157 000274 016705 000000G ALOCDB: MOV $SESDB,R5 ; Set up database descriptor
158 000300 016501 000036 MOV N$LVC(R5),R1 ; Get max # of logical links allowed
159 000304 005201 INC R1 ; Compute # of bytes to allocate
160 000306 006301 ASL R1 ;
161 000310 CALL @ALOCB ; Allocate the block
162 000314 103416 BCS 100$; If CS, allocation failure
163 000316 010065 000040 MOV R0,N$LVC+2(R5) ; Store address of LLT table
164 000322 006201 ASR R1 ; # of words in table
165 000324 005020 10$: CLR (R0)+ ; Clear out the table
166 000326 SOB R1,10$; ...
167
168 000332 005065 000032 CLR N$ACTL(R5) ; No active logical links yet
169 000336 005065 000050 CLR N$MBXQ(R5) ; Initialise mailbox queue
170 000342 005065 000052 CLR N$GENQ(R5) ; Initialise general delivery queue listhead
171 000346 000241 CLC ; Indicate success
172 000350 RFTURN
173
174 000352 100$: CALL DEALDB ; Deallocate databases
175 000356 000261 SEC ; Indicate failure
176 000360 RETURN

```

SESMN - Session control main Lo MACRO V05.03b Friday 28-Jun-85 19:57<sup>F 3</sup>  
Table of contents

|     |     |                            |
|-----|-----|----------------------------|
| 6-  | 42  | Macro definitions          |
| 7-  | 59  | Define local macros        |
| 8-  | 66  | Session control main logic |
| 9-  | 126 | Execute a directive        |
| 10- | 153 | Local data areas           |

| SYMBOL  | VALUE    | REFERENCES               |
|---------|----------|--------------------------|
| N\$FLG  | 000005   | 8-94 *8-105 8-116 *8-122 |
| N\$MBXQ | 000050   | 8-99                     |
| N\$ECL  | = *****  | 8-118 8-121              |
| N\$EVL  | = 000001 | #4-2 8-118 8-121         |
| N\$SES  | = 000001 | #6-57 8-118 8-121        |
| N\$SVCT | = *****  | 8-103 8-114              |
| OP\$INI | = 000000 | #6-54                    |
| OP\$TER | = 000001 | #6-54                    |
| PH\$HDE | = 000004 | #6-54                    |
| PH\$LOC | = 000002 | #6-54                    |
| PH\$MTS | = 000003 | #6-54                    |
| PH\$UMP | = 000000 | #6-54                    |
| PH\$WCS | = 000001 | #6-54                    |
| RCP1    | 000274 R | 8-89 #10-155             |
| RCP2    | 000312 R | 8-90 #10-156             |
| RE\$ADC | = 000004 | #6-54                    |
| RE\$ADF | = 000017 | #6-54                    |
| RE\$ADR | = 000007 | #6-54                    |
| RE\$BLK | = 000010 | #6-54                    |
| RE\$CAF | = 000014 | #6-54                    |
| RE\$DAT | = 000001 | #6-54                    |
| RE\$DRP | = 000016 | #6-54                    |
| RE\$LDT | = 000013 | #6-54                    |
| RE\$LSN | = 000012 | #6-54                    |
| RE\$NML | = 000001 | #6-54                    |
| RE\$OPE | = 000004 | #6-54                    |
| RE\$OPR | = 000000 | #6-54 8-118 8-121        |
| RE\$RCV | = 000001 | #6-54                    |
| RE\$SED | = 000011 | #6-54                    |
| RE\$SKW | = 000006 | #6-54                    |
| RE\$STA | = 000002 | #6-54                    |
| RE\$SUM | = 000003 | #6-54                    |
| RE\$SYN | = 000000 | #6-54                    |
| RE\$TME | = 000021 | #6-54                    |
| RE\$TMO | = 000000 | #6-54                    |
| RE\$TMR | = 000020 | #6-54                    |
| RE\$UPT | = 000002 | #6-54                    |
| RE\$URE | = 000003 | #6-54                    |
| RE\$VER | = 000005 | #6-54                    |
| RE\$VRQ | = 000015 | #6-54                    |
| RT\$INI | = 000002 | #6-54                    |
| RT\$OFF | = 000001 | #6-54                    |
| RT\$ON  | = 000000 | #6-54                    |
| R\$MPL  | = *****  | 6-48 8-77 10-158         |
| R.Q\$TN | = 000002 | 10-156                   |
| SC\$OFF | = 000001 | #6-54 8-118 8-121        |
| SC\$ON  | = 000000 | #6-54 8-121              |
| SC\$RST | = 000003 | #6-54                    |
| SC\$SHU | = 000002 | #6-54 8-118              |
| SE\$INI | = *****  | 8-85                     |
| SV\$DUM | = 000001 | #6-54                    |
| SV\$LOA | = 000000 | #6-54                    |

GX

```

332
333 000730 112713 000010 30$: MOVB #ST$DAT,(R3) ; Move link into data state
334 000734 017700 000000G MOV @DECPT,R0 ; Get address of DEC home block
335 000740 116063 000044 000065 MOVB D$INAC(R0),L.LPT(R3) ; Store inactivity timer
336
337 000746 026563 000016 000004 90$: CMP N$SLA(R5),L.RLA(R3)
338 000754 001013 BNE 110$; Send 'NO LINK' if message received for idle link
339 000756 122713 000010 CMPB #ST$DAT,(R3) ; Is link in data state?
340 000762 001007 BNE 100$; If NE, no ... toss the message
341
342 000764 052763 100000 000024 BIS #LA.ACK,L.USA(R3)
343 000772 CALLE FRCACK ; Acknowledge the CC
344
345 001002 100$: RETURN
346
347 001004 110$: CALLE NOLINK ; Send 'NO LINK' DC message
348 001014 005067 000000G CLR $RCCB ; We have re-used the CCB
349 001020 RETURN

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                                                     |
|---------|------------|----------------------------------------------------------------|
| FNDDL Y | = ***** GX | 11-316                                                         |
| FNDLLT  | = ***** GX | 11-302 12-368 13-419                                           |
| FRCACK  | = ***** GX | 11-343                                                         |
| GETOPT  | = 001320 R | 11-299 12-365 #14-470                                          |
| GETSDB  | = ***** GX | 16-559                                                         |
| KILLNK  | = ***** GX | 12-386 13-435                                                  |
| KISAR6  | = ***** GX | *8-101 *9-186 *18-621                                          |
| LA.ACK  | = 100000   | 11-342                                                         |
| LF.HMF  | = 000040   | 15-527                                                         |
| LF.HSF  | = 000020   | 15-523                                                         |
| LT.CCA  | = 000020   | 15-516                                                         |
| LT.LCL  | = 000001   | 15-504                                                         |
| LT.LPL  | = 000002   | 15-507                                                         |
| L.CSTA  | = 000037   | *16-574 *17-595                                                |
| L.DCR   | = 000100   | *9-228 *12-373 *13-430 16-558                                  |
| L.FLAG  | = 000014   | *15-523 *15-527                                                |
| L.LLA   | = 000002   | 10-265                                                         |
| L.LPT   | = 000065   | *11-335                                                        |
| L.NXTH  | = 000010   | *15-509                                                        |
| L.REM   | = 000006   | 9-182 *15-499 *15-508                                          |
| L.RLA   | = 000004   | 9-180 *11-314 11-337 13-423 *15-519                            |
| L.RTQ   | = 000060   | 18-612 *18-615                                                 |
| L.RTYD  | = 000055   | *16-573 *17-593                                                |
| L.SEGZ  | = 000076   | *15-534                                                        |
| L.TMRD  | = 000054   | *11-315 *13-434 *16-571 *18-625                                |
| L.USA   | = 000024   | *11-342                                                        |
| L.VER   | = 000015   | *15-512                                                        |
| MC.CI   | = 000020   | 9-170                                                          |
| MC.DI   | = 000060   | 16-561                                                         |
| MC.RC   | = 000140   | 18-622                                                         |
| MF.CTL  | = 000010   | 9-170 16-561 18-622                                            |
| NOLINK  | = ***** GX | 11-347                                                         |
| NOOP    | = 000076 R | #8-121 8-127 8-132 8-134                                       |
| NS\$DON | = 000000   | 12-398 12-399 12-401 12-403 13-443 13-444 13-446 13-447 13-448 |
| NS\$WDC | = 000004   | 17-595                                                         |
| NT\$CON | = 000000   | 16-574                                                         |
| NT\$DIS | = 000014   | 18-619                                                         |
| NSDLA   | = 000020   | 16-560                                                         |
| NSDLY   | = 000014   | *8-109 *8-110 9-154 *9-234                                     |
| NSERRC  | = 000022   | 16-571 18-625                                                  |
| NSLVC   | = 000036   | 9-224 9-228 *12-374                                            |
| NSPLLT  | = 000030   | 9-174 9-175 10-256 10-259                                      |
| NSSLA   | = 000016   | 9-205                                                          |
| NSNOD   | = 000012   | *8-111 *8-112 9-180 10-254 10-263 11-314 11-337 13-423 15-519  |
| NS\$ECL | = *****    | *8-100 9-182 15-499 15-501                                     |
| NS\$EVL | = 000001   | 9-197 9-198 11-310 11-311 12-380 12-381 12-382 13-422          |
| NS\$NCT | = 000001   | #4-2 9-197 9-198 9-198 11-308 11-310 11-310 11-311 11-311      |
| NS\$SES | = 000001   | 12-378 12-380 12-381 12-381 12-382 12-382 12-382 13-422        |
| NS\$SLI | = *****    | #6-55 9-197 9-198 11-310 11-311 12-380 12-381 12-382 13-422    |
| NS\$VCT | = *****    | 9-191 9-209 11-322 15-536 9-237 10-264 11-302 11-316 11-320    |
|         |            | 8-101 8-118 9-186 9-216                                        |

```

220 .SBTTL Notify user of logical link disconnect/abort
221
222 ;+
223 ;**--USRDIS-Notify user of logical link disconnect or abort
224 ;
225 ; Queue a CCB to the user's mailbox indicating that the logical link
226 ; has been disconnected or aborted.
227 ;
228 ; Inputs:
229 ; R3 = Virtual address of LLT
230 ; R5 = Address of database descriptor
231 ;
232 ; Registers modified:
233 ; R0, R1, R2, R4
234
235 000364 016300 000040 USRDIS: MOV L,WIND(R3),R0 ; Check for deallocated window block
236 000370 001452 BEQ 20$; If EQ, window block is already gone
237
238 ; IF DF N$$BUF
239 TSTB W,KAST(P0) ; Are there any kernel ASTs outstanding
240 BNE 20$; If NE, yes - issue disconnect later
241 .ENDC
242
243 000372 CALL @CCBG1 ; Allocate a CCB
244 000376 103447 BCS 20$; If CS, allocation failure
245
246 000400 112763 000010 000036 MOVB #US$WDS,L,USTA(R3)
247 000406 012700 000003 MOV #NT.DSC,R0 ; Assume user disconnect
248 000412 016364 000100 000004 MOV L,DCR(R3),C,NSP(R4)
249 000420 001410 BEQ 10$; If EQ, user disconnect
250 000422 012700 000004 MOV #NT.ABT,R0 ; Assume user abort
251 000426 026327 000100 000011 CMP L,DCR(R3),#ER$ABT
252 000434 001402 BEQ 10$; If EQ, user abort
253 000436 012700 000005 MOV #NT.ABO,R0 ; else network abort
254
255 000442 110064 000010 10$: MOVB R0,C,FNC(R4) ; Set up CCB for disconnect type and reason
256 000446 CALL PNTCCB ; Point CCB at optional data
257
258 000452 016303 000040 MOV L,WIND(R3),R3 ; Get address of window block
259 000456 010364 000012 MOV R3,C,STS(R4) ; and save for completion later
260 000462 152763 000010 000002 BISB #WS.DIP,W,STAT(R3)
261 000470 116364 000003 000011 MOVB W,LUN(R3),C,MOD(R4)
262 000476 016367 000012 000000G MOV W,MBOX(R3),$MA:BX
263 000504 CALL ADDMAI ; Add mail to user mailbox
264
265 000510 010304 MOV R3,R4 ; Copy window block address
266 000512 CALL FLSHIO ; Flush pending I/O packets
267
268 000516 20$: RETURN

```

```

59 .IF DF N$$$SL1
60 :SBTTL System level interface request - transmit enable
61 ;+
62 **-.SEXME-System level interface request - transmit enable
63 :
64 This routine processes a single request for system level interface
65 service via the transmit enable entry point.
66 :-
67 Inputs:
68 R3 = Subfunction code
69 R4 = Address of CCB
70 R5 = Address of database descriptor
71 :
72 .PSECT $HIGH
73
74 .SEXME::CALLR @XMETBL(R3) ; Dispatch to processing routine
75
76 ;+
77 ; Transmit enable dispatch table
78 :-
79 XMETBL: .WORD SLICON ; Connect request
80 .WORD SLIACC ; Accept connection
81 .WORD SLIREJ ; Reject connection
82 .WORD .+1 ; Data segment (illegal)
83 .WORD .+1 ; Data message request (illegal)
84 .WORD .+1 ; Data segment request (illegal)
85 .WORD .+1 ; Interrupt message (illegal)
86 .WORD .+1 ; Interrupt request (illegal)
87 .WORD SLIDSL ; Disconnect
88 .WORD SLIABO ; Abort
89 .WORD SLIGLN ; Get local node info

```



```

552 .SBTTL Copy node name to image field buffer
553 ;+
554 ;**--NODIMG-Copy node name to image field
555 ;
556 ; copy a node name (6 bytes, space filled) to an image output field.
557 ;
558 ; Inputs:
559 ; R0 = Address of next available byte in the output buffer
560 ; R3 = Address of node name
561 ;
562 ; Outputs:
563 ; R0 = Address of next available byte in the output buffer
564 ;
565 ; Registers modified:
566 ; R1, R3
567 ;
568 .PSECT
569
570 NODIMG: SAVRG <R2> ; Get some free registers
571 CLRB (R0)+ ; Initialise count
572 MOV R0,-(SP) ; Save pointer to start of node name
573 MOV #6,R2 ; Set max length of node name
574
575 10$: MOVB (R3)+,R1 ; Get next character from node name
576 CMPB R1,#' ' ; Space terminates the node name
577 BEQ 20$; If EQ, all done
578 MOVB R1,(R0)+ ; Plant character in output buffer
579 SOB R2,10$; Scan max of six characters
580
581 20$: MOV R0,R1 ; Copy output address
582 MOV (SP)+,R2 ; Recover pointer to start of node name
583 SUB R2,R1 ; Compute length of node name
584 MOVB R1,-(R2) ; Plant count in buffer
585 RESRG <R1> ; Restore registers
586 RETURN

```

SESSL1 - Session control system MACRO V05.03b Friday 28-Jun-85 19:58 Page 28-3  
Symbol table

Z.NAM 000004 Z.SCH 000007 \$\$\$HFT= 000001 \$\$\$ = 000062 .\$\$\$\$.= 000034  
Z.PCB 000012

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000000 001 (RW,I,LCL,REL,CON)

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 79  
Work file writes: 104  
Size of work file: 27656 Words ( 109 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:30.33

SY:SESSL111S.V2,[131,134]SESSL111S/CR/~SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCS/PA:1,[131,10]V2,SESSL1

```

336
337 000664 016564 000012 000002 50$: MOV N$SNOD(R5),E$NNOD(R4)
338 000672 105264 000004 000004 INCB E$NUSE(R4) ; Update usage count
339 000676 126464 000040 000004 CMPB E$NML(R4),E$NUSE(R4)
340 000704 103003 60$: BHS 60$; Check for high water mark exceeded
341 000706 116464 000004 000040 MOVB E$NUSE(R4),E$NML(R4)
342
343 000714 116465 000005 000014 60$: MOVB E$NRTP(R4),N$DLY(R5)
344 000722 005727 TST (PC)+ ; Indicate database allocated
345 000724 000261 100$: SEC ; Indicate failed to allocate database
346
347 000726 MAPLLT ; Recover mapping to LLT
348 000734 010463 000074 MOV R4,L.CTR(R3) ; Set up pointer to ECL node database (or zero)
349 000740 RESRG <R4>
350 000742 RETURN

```

```

789 .SBTTL Deallocate ECL node database
790 +
791 ***-DEADB-Deallocate ECL node database
792 Deallocate the ECL node database for this logical link.
793
794 -
795 Inputs:
796 R3 = Virtual address of LLT
797 R5 = Address of database descriptor
798
799 Registers modified:
800 R0, R1
801
802 DEADB: SAVRG <R3> ; Get a free register
803 MOV L.CTR(R3),-(SP) ; Get address of counter block
804 MOV L.LLA(R3),R3 ; and the LLA
805
806 MAP N$ENC+4(R5) ; Map to the ECL databases
807 MOV R5,R0 ; Compute address of ECL database listhead
808 ADD #N$ENC,R0 ; ...
809
810 10$: MOV R0,R1 ; Save address of previous block
811 MOV (R0),R0 ; Get next database
812 BEQ 30$; If EQ, no more
813 CMP R0,(SP) ; Is this the one we're looking for?
814 BNE 10$; If NE, no ... keep looking
815
816 CMP R3,E$NLLA(R0) ; Are we timing this link?
817 BNE 20$; If NE, no
818 MOV #-1,E$STRT(R0) ; Stop the round trip timer
819
820 20$: DECB E$NUSE(R0) ; Reduce usage count
821 BNE 30$; If NE, more active users
822
823 MOV #-1,E$STRT(R0) ; Stop the round trip timer
824 CLR E$NLLA(R0) ; and logical link address
825 CMP R0,N$ENC+2(R5) ; Is this block already at the end of the list?
826 BEQ 30$; If EQ, yes
827
828 MOV (R0),(R1) ; Unlink block from list
829 CLR (R0) ; Clear out link word
830 MOV R0,#N$ENC+2(R5) ; Relink at end of list
831 MOV R0,N$ENC+2(R5) ; ...
832
833 30$: TST (SP)+ ; Clean up stack
834 RESRG <R3> ; Restore register
835 RETURN

```

SESSUB - Session control subrou MACRO V05.03b Friday 28-Jun-85 19:59 Page 32  
Process image field

```

1272 .SBTTL Process image field
1273 :+
1274 : **--PROIMG-Process image field
1275 :
1276 : Process an image mode field from a connect initiate message.
1277 :-
1278 : Inputs:
1279 : R0 = Pointer into connect pending block
1280 : R1 = Maximum length of image field allowed
1281 : R2 = Pointer to start of image field
1282 :
1283 : Outputs:
1284 : R2 = Moved past image field
1285 : 'C' Clear - Valid image field processed
1286 : 'C' Set - Format error in image field length
1287 :
1288 : Registers modified:
1289 : R1, R5
1290
1291 PROIMG: MOVB (R2)+,R5 ; Get length of image field
1292 CMP R1,R5 ; Is the field too long?
1293 BLO 20$; If LO, yes ... C-bit is set
1294
1295 MOV R5,(R0)+ ; Fill in byte count
1296 BEQ 20$; If EQ, null
1297 10$: MOVB (R2)+,(R0)+ ; Copy image field
1298 SOB R5,10$; ...
1299
1300 20$: RETURN

```

SESSUB - Session control subrou MACRO V05.03b Friday 28-Jun-85 19:59 Page 33

SESSUB - Session control subrou MACRO V05.03b Friday 28-Jun-85 19:59 Page 42-1  
Symbol table

|                  |                    |                  |                  |                  |
|------------------|--------------------|------------------|------------------|------------------|
| ACCLLT= ***** GX | CEMUL = ***** GX   | CPYOPT 001702RG  | C.BUF 000014     | D\$LNAM 000006   |
| AC\$DNT= 000002  | CE.ABO= 100362     | CP.DCF= 000040   | C.BUF1 000014    | D\$LNUM 000014   |
| AC\$X25= 000001  | CE.DAO= 100346     | CP.HDL= 000007   | C.BUF2 000024    | D\$LST 000047    |
| ADDGNG 000000RG  | CE.DIS= 100366     | CP.PS = 177400   | C.CNT 000020     | D\$MAXC 000064   |
| ADDLNK 000056RG  | CE.ERR= 100370     | CP.PSI= 000200   | C.CNT1 000020    | D\$MAXH 000066   |
| ADDMAI 000356RG  | CE.ILN= 100350     | CP.XCF= 000100   | C.CNT2 000030    | D\$MAXV 000070   |
| ADDOPT 000426RG  | CE.LTO= 100356     | CP.2FR= 000030   | C.FLG 000022     | D\$MILL 000040   |
| AES\$CIR= 000003 | CE.MOP= 100372     | CSBGT = ***** GX | C.FLG1 000022    | D\$MNOD 000041   |
| AESLIN= 000001   | CE.NTE= 100361     | CSBRT = ***** GX | C.FLG2 000032    | D\$NA 000062     |
| AESMOD= 000004   | CE.RTE= 100376     | CS.ABO= 000100   | C.FNC 000010     | D\$NBEA 000056   |
| ALLDB 000524RG   | CE.SRC= 100364     | CS.BRO= 000002   | C.LIN 000006     | D\$NBRA 000054   |
| ALOCB = ***** GX | CE.STP= 100352     | CS.BUF= 000200   | C.LNK 000000     | D\$NEND= 000054  |
| ASSCHK= 000000   | CE.TME= 100354     | CS.CES= 000002   | C.MOD 000011     | D\$NLN 000030    |
| ASSCPS= 000000   | CE.TMQ= 100374     | CS.CHN= 000010   | C.NSP 000004     | D\$NN 000060     |
| ASSPRI= 000000   | CE.UNS= 100344     | CS.CMP= 000200   | C.PRO 000042     | D\$OUTT 000043   |
| ASSTRP= 000000   | CF.CHN= 000001     | CS.DCR= 000400   | C.RSV 000002     | D\$RETF 000050   |
| BRKLNK 000744RG  | CF.EOM= 000004     | CS.DEF= 000004   | C.STA 000007     | D\$RNN 000002    |
| BYTE3 = 000050   | CF.HDR= 000020     | CS.DEV= 000002   | C.STS 000012     | D\$RTMR 000076   |
| CAT5 000774RG    | CF.LB = 100000     | CS.DIS= 000040   | C.URM 177776     | D\$SEG 000036    |
| CB.ACN 000114    | CF.LIN= 000002     | CS.ENA= 000001   | C.XACP 000004    | D\$SER 000032    |
| CB.ACT 000112    | CF.SOM= 000010     | CS.ENB= 000020   | C.XID 000035     | D\$SQRL 000052   |
| CB.CCB= 000002   | CF.SYN= 000040     | CS.ERR= 100000   | C.XLEN 000044    | D\$SUBG= 177514  |
| CB.DDM= 000040   | CF.TRN= 000100     | CS.FTL= 001000   | C.XPLI 000040    | D\$SISK= 000000  |
| CB.DFM 000006    | CL\$ASZ= 010500    | CS.HCR= 000001   | C.XPT 000034     | D\$SL11= 000001  |
| CB.DGR 000010    | CL\$DLL= 000500    | CS.HFE= 002000   | C.XSVC 000042    | D\$SYNC= 000000  |
| CB.DLC= 000020   | CL\$ECL= 000300    | CS.LST= 040000   | C.XTC 000037     | D\$SYNM= 000000  |
| CB.DL1 000010    | CL\$LDN= 010400    | CS.MTL= 004000   | C.X25 000036     | EF\$ACT= 000001  |
| CB.DL2 000014    | CL\$MAN= 000000    | CS.RNG= 000010   | CSTA 000000RG    | ERSABO= 000010   |
| CB.DOB 000007    | CL\$MFL= 000010    | CS.ROV= 000004   | DEACB = ***** GX | ERSABD= 000046   |
| CB.DR1 000012    | CL\$PAZ= 034100    | CS.RSN= 010000   | DEADB 001754R    | ERSABT= 000011   |
| CB.DR2 000016    | CL\$PLH= 034000    | CS.SHU= 000001   | DECPY = ***** GX | ERSACC= 000042   |
| CB.DUS 000012    | CL\$PLL= 000600    | CS.SID= 000002   | DL\$AST= 000002  | ERSCDI= 000052   |
| CB.LGT 000156    | CL\$PRT= 034200    | CS.STR= 000004   | DL\$HLT= 000000  | ERSCOM= 000047   |
| CB.NOD 000000    | CL\$ROU= 010000    | CS.SUC= 000001   | DL\$IST= 000001  | ERSFMT= 000005   |
| CB.OPD 000134    | CL\$SES= 000200    | CS.TMO= 020000   | DL\$MAI= 000004  | ERSMLB= 000006   |
| CB.OPT 000136    | CL\$SFL= 000004    | CS.XUR= 000004   | DL\$OFF= 000001  | ERSNNF= 000012   |
| CB.PSL 000100    | CL\$SGE= 035000    | CTRSES= ***** GX | DL\$ON = 00000   | ERSNOD= 000002   |
| CB.PSW 000102    | CL\$SSE= 035100    | CVADAS 000104R   | DL\$RUN= 000003  | ERSNSL= 000013   |
| CB.RDB= 000004   | CL\$TRN= 000400    | CVT 000144R      | DL\$SHU= 000002  | ERSNSR= 000003   |
| CB.RQD 000056    | CL\$TYP= 000001    | CVTADD 000210R   | DL\$SYN= 000005  | ERSRES= 000001   |
| CB.RQ1 000060    | CL\$XL2= 013700    | CVTARE 000202R   | DONE 003506R     | ERSSTA= 000051   |
| CB.SDB= 000010   | CL\$XL3= 013600    | CVTBL 001144R    | DSTATE 000760R   | ERSUOB= 000004   |
| CB.SFM 000032    | CL\$X25= 013500    | CVTC 000030R     | D\$AMX 000072    | EVDC = ***** GX  |
| CB.SGR 000034    | CL.MU1= 000001     | CVSMASK= 000003  | D\$AMXH 000074   | EVLSES= ***** GX |
| CB.SLI= 000100   | CL.MU2= 000002     | CV\$31 = 000001  | D\$ANN 000000    | EVSACF= 000201   |
| CB.SL1 000034    | CL.RES= 177774     | CV\$32 = 000000  | D\$BRPR 000102   | EVSADR= 000420   |
| CB.SL2 000040    | CM\$PINT= ***** GX | CV\$40 = 000002  | D\$BRTM 000100   | EVSADU= 000417   |
| CB.SOB 000033    | CM.CIR= 000002     | CV\$4Q = 000001  | D\$DELF 000045   | EVSAPL= 000400   |
| CB.SR1 000036    | CM.CON= 000200     | CV\$REM= 000020  | D\$DELW 000046   | EVSARC= 000421   |
| CB.SR2 000042    | CM.FMT= 100000     | CV\$REQ= 000002  | D\$END = 000104  | EVSauc= 000010   |
| CB.SUS 000036    | CM.HRD= 000002     | CV\$RUI= 000040  | D\$FNB 000034    | EVSaus= 000003   |
| CB.XLB= 000001   | CM.LIN= 000000     | CV\$SMC= 000010  | D\$HIOR 000024   | EVScof= 000520   |
| CCBGT = ***** GX | CM.LOD= 000001     | CV\$UNL= 000004  | D\$HOST 000022   | EVScoz= 000011   |
| CCBRT = ***** GX | CM.XLO= 000004     | CS\$ORE= 000400  | D\$INAC 000044   | EVSDBR= 000302   |
| CC.LLC= 000200   | CNTRL 001134R      | CS\$RSH= 177564  | D\$INCT 000042   | EVSgas= 035101   |
| CEACC = ***** GX | CNVCI 001152RG     | C.ADD 000034     | D\$IPL 000051    | EVSHCe= 035114   |
| CEDIV = ***** GX | CPYIMG 001662R     | C.BID 000003     | D\$LID 000020    | EVSHCi= 035113   |

SESSUB11S CREATED BY MACRO ON 28-JUN-85 AT 20:00 PAGE 10 F 15  
 MACRO CROSS REFERENCE CREF 04.00

| MACRO NAME | REFERENCES                                                                                                                                                                                                                                                                                                                                 |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CALL       | 8-136 8-167 13-399 13-401 13-403 14-432 15-490 16-565 17-621 17-627<br>17-646 17-661 17-667 17-672 17-716 17-718 18-752 21-852 21-861 21-867<br>21-873 22-907 23-952 27-1105 27-1111 29-1190 30-1215 33-1344 34-1366 34-1367<br>35-1391 35-1409 35-1418 36-1455 37-1486 37-1487 38-1527 38-1533 40-1613 41-1634<br>41-1637 41-1645 42-1709 |
| CALLE      | #6-44 9-238 34-1365 35-1404 40-1602                                                                                                                                                                                                                                                                                                        |
| CALLR      | 12-368 23-962 24-1005 24-1014                                                                                                                                                                                                                                                                                                              |
| CALLX      | #6-44 8-131 8-198 #9-238 9-238 #34-1365 34-1365 #35-1404 35-1404 35-1425<br>#40-1602 40-1602                                                                                                                                                                                                                                               |
| CCBDFS     | #6-46 6-50                                                                                                                                                                                                                                                                                                                                 |
| CNBDFF     | #6-47 6-56                                                                                                                                                                                                                                                                                                                                 |
| COUNTS     | #6-44 8-206 10-263 17-706                                                                                                                                                                                                                                                                                                                  |
| CTRDFS     | #6-46 6-53                                                                                                                                                                                                                                                                                                                                 |
| DHBDFF     | #6-48 6-63                                                                                                                                                                                                                                                                                                                                 |
| ECDDBS     | #6-46 6-51                                                                                                                                                                                                                                                                                                                                 |
| EVLDFS     | #6-46 6-54                                                                                                                                                                                                                                                                                                                                 |
| EVT\$      | #6-44 11-326                                                                                                                                                                                                                                                                                                                               |
| LLTDFS     | #6-46 6-55                                                                                                                                                                                                                                                                                                                                 |
| LLWDFS     | #6-47 6-58                                                                                                                                                                                                                                                                                                                                 |
| MAP        | #6-44 10-259 10-266 11-298 11-347 17-664 19-780 19-786 20-806 27-1108<br>35-1387 42-1711                                                                                                                                                                                                                                                   |
| MAPLLT     | #6-45 10-259 11-347 19-786 35-1387 42-1711                                                                                                                                                                                                                                                                                                 |
| MBXDFS     | #6-46 6-52                                                                                                                                                                                                                                                                                                                                 |
| MSGDFS     | #6-47 6-59                                                                                                                                                                                                                                                                                                                                 |
| NSSYMS     | #6-47 6-61                                                                                                                                                                                                                                                                                                                                 |
| OBJDFS     | #6-47 6-60                                                                                                                                                                                                                                                                                                                                 |
| RECMAP     | #6-45 17-660 21-851 23-951                                                                                                                                                                                                                                                                                                                 |
| RESMAP     | #6-45 17-720 21-853 23-956 30-1237 34-1368                                                                                                                                                                                                                                                                                                 |
| RESRG      | #6-44 8-190 8-208 9-240 10-276 11-349 15-473 15-498 16-567 16-576<br>17-628 17-721 19-785 20-834 21-875 23-953 23-957 26-1075 33-1342 34-1369<br>36-1457 37-1488 38-1535 40-1617 42-1710                                                                                                                                                   |
| RETURN     | 7-96 8-191 8-209 9-241 10-277 11-350 13-408 14-449 15-474 16-577<br>17-722 20-835 21-854 21-876 22-910 23-958 26-1076 28-1157 29-1191 30-1238<br>31-1270 32-1300 33-1345 34-1370 35-1435 36-1458 37-1489 38-1536 39-1560 40-1618<br>41-1648 42-1712                                                                                        |
| RNBDFS     | #6-47 6-57                                                                                                                                                                                                                                                                                                                                 |
| SAVMAP     | #6-45 17-609 21-850 23-929 30-1211 34-1363                                                                                                                                                                                                                                                                                                 |
| SAVRG      | #6-44 8-115 9-224 10-258 11-297 15-465 15-476 16-528 16-561 17-608<br>17-625 19-772 20-802 21-857 23-928 23-949 26-1063 33-1338 34-1362 36-1446<br>37-1483 38-1505 40-1583 42-1698                                                                                                                                                         |
| SLIDFS     | #6-48 6-62                                                                                                                                                                                                                                                                                                                                 |
| SOB        | 8-123 8-156 11-331 15-497 17-638 19-783 32-1298 33-1341 39-1558                                                                                                                                                                                                                                                                            |

```

53 .SBTTL Session control timer support
54 ;+
55 ;**--SESTIM-Session control timer support
56 ;
57 ;this routine is called to provide timer support for logical links
58 ;in the connect and disconnect states.
59 ;
60 ;Inputs:
61 ;R5 = Address of database descriptor
62 ;
63 ;Registers modified:
64 ;R0, R1, R2, R3, R4
65
66 SESTIM: MOV N$LVC(R5),R1 ; Get # of LLT's to scan
67 MOV N$LVC+2(R5),R2 ; Get pointer to logical link table
68 TST (R2)+ ; Skip over first entry
69
70 10$: MOV (R2)+,R3 ; Get address of next LLT
71 BEQ 20$; If EQ, none present
72 CALLE ACCLLT ; Gain access to this LLT
73
74 MOVB (R3),R0 ; Get the logical link state
75
76 CMP R0,#SI$DAT ; Is it in normal data transfer state?
77 BEQ 20$; If EQ, yes ... ECL handles timing
78 BISB #NFTIM,NFLG(R5) ; Make sure timer is still running
79
80 SAVRG <R1,R2> ; Save scan variables
81 MOV R3,R2 ; Compute address of data subchannel timer
82 ADD #L1MRD,R2 ;
83 CALL @TIMDSP-2(R0) ; Dispatch to timer processing routine
84 RESRG <R2,R1> ; Recover scan variables
85
86 20$: SOB R1,10$; Scan all logical links
87 TIMNOP: RETURN
88
89 ;+
90 ;Timer support dispatch table
91 ;
92 ;
93 TIMDSP: .WORD TIMCIS ; CI sent
94 .WORD TIMCC ; Received CI, CC sent
95 .WORD TIMCIR ; Received CI
96 .WORD +1 ; Normal data transfer
97 .WORD TIMDIP ; Disconnect in progress
98
99 .IF DF N$$SLI
100 .WORD SLIDCT ; Disconnect pending (SLI)
101
102 .IFF
103
104 .WORD TIMNOP ; Disconnect pending
105
106 .ENDC
107

```



|                     |                 |                  |                 |                  |
|---------------------|-----------------|------------------|-----------------|------------------|
| ACPIDL 000000RG     | CP.2FR= 000030  | C.RSV 000002     | ER\$ACC= 000042 | FS.WLB= 001000   |
| AUXTSK 000164R 002  | CS.ABO= 000100  | C.STA 000007     | ER\$CDI= 000052 | FS.XKL= 002000   |
| AS\$CHK= 000000     | CS.BRO= 000002  | C.STS 000012     | ER\$COM= 000047 | FS.XOF= 010000   |
| AS\$CPS= 000000     | CS.BUF= 000200  | C.URM 177776     | ER\$FMT= 000005 | FS.XON= 007000   |
| AS\$PRI= 000000     | CS.CES= 000002  | C.XACP 000004    | ER\$MLB= 000006 | FS.ZER= 002000   |
| AS\$TRP= 000000     | CS.CHN= 000010  | C.XID 000035     | ER\$NNF= 000012 | FS\$LV= 000001   |
| CB.CCB= 000002      | CS.CMP= 000200  | C.XLEN 000044    | ER\$NOD= 000002 | GS\$TPP= 000000  |
| CB.DDM= 000040      | CS.DCR= 000400  | C.XPLI 000040    | ER\$NSL= 000013 | GS\$TSS= 000000  |
| CB.DLC= 000020      | CS.DEF= 000004  | C.XPT 000034     | ER\$NSR= 000003 | GS\$TTK= 000000  |
| CB.RDB= 000034      | CS.DEV= 000002  | C.XSVC 000042    | ER\$RES= 000001 | GS\$WRD= 000000  |
| CB.SDB= 000010      | CS.DIS= 000040  | C.XTC 000037     | ER\$STA= 000051 | IN.DAT= 000400   |
| CB.SLI= 000100      | CS.ENA= 000001  | C.X25 000036     | ER\$UOB= 000004 | IN.ILS= 000001   |
| CB.XLB= 000001      | CS.ENB= 000020  | DECTP = ***** GX | ER\$XPR= 000000 | IODUN 000466RG   |
| CCBDSP 000000OR 002 | CS.ERR= 100000  | D\$AMXC 000072   | FC.CCP= 000020  | IODUN1 000464RG  |
| CCBRT = ***** GX    | CS.FTL= 001000  | D\$AMXH 000074   | FC.CTL= 000006  | IOERR 000454RG   |
| CC.LLC= 000200      | CS.HCR= 000001  | D\$ANN 000000    | FC.KCP= 000016  | IOFIN = ***** GX |
| CE.ABO= 000362      | CS.HFE= 002000  | D\$BRPR 000102   | FC.KIL= 000004  | IOSUC 000460RG   |
| CE.DAO= 100346      | CS.LST= 040000  | D\$BRTM 000100   | FC.MAN= 000024  | IS.SUC= ***** GX |
| CE.DIC= 100366      | CS.MTL= 004000  | D\$DELF 000045   | FC.MLD= 000026  | IS\$RAR= 000000  |
| CE.ERR= 100370      | CS.RNG= 000010  | D\$DELF 000046   | FC.PCT= 000030  | IS\$RDN= 000000  |
| CE.ILN= 100350      | CS.ROV= 000004  | D\$END = 000104  | FC.PWR= 000022  | I.FCN = ***** GX |
| CE.LTO= 100356      | CS.RSN= 010000  | D\$FNB 000034    | FC.RCE= 000002  | I.LN2 = ***** GX |
| CE.MOP= 100372      | CS.SHU= 000001  | D\$HIOR 000024   | FC.RCP= 000014  | I.PRM = ***** GX |
| CE.NTE= 100361      | CS.SID= 000002  | D\$HOST 000022   | FC.TIM= 000010  | I.TCB = ***** GX |
| CE.RTE= 100376      | CS.STR= 000004  | D\$INAC 000044   | FC.XCP= 000012  | KISAR6= ***** GX |
| CE.SRC= 100364      | CS.SUC= 000001  | D\$INCT 000042   | FC.XME= 000000  | K\$CNT= 177546   |
| CE.STP= 100352      | CS.TMO= 020000  | D\$IPL 000051    | FS.AST= 000000  | K\$CSR= 177546   |
| CE.TME= 100354      | CS.XUR= 000004  | D\$LID 000020    | FS.CIB= 002000  | K\$LDLC= 000000  |
| CE.TMO= 100374      | CV\$MSK= 000003 | D\$LNAM 000006   | FS.CRA= 001000  | K\$TPS= 000074   |
| CE.UNS= 100344      | CV\$31 = 000001 | D\$LNAM 000014   | FS.DIS= 013000  | LA.ACK= 100000   |
| CF.CHN= 000001      | CV\$32 = 000000 | D\$LST 000047    | FS.DVC= 001000  | LA.CRS= 020000   |
| CF.EOM= 000004      | CV\$40 = 000002 | D\$MAXC 000064   | FS.ENB= 012000  | LA.MSK= 170000   |
| CF.HDR= 000020      | CX.GDQ= 000001  | D\$MAXH 000066   | FS.EXI= 001000  | LA.NAK= 110000   |
| CF.LB = 100000      | CX.REM= 000020  | D\$MAXV 000070   | FS.GET= 006000  | LA.NMS= 010000   |
| CF.LIN= 000002      | CX.REQ= 000002  | D\$MLL 000040    | FS.HLT= 000000  | LA.RES= 040000   |
| CF.SOM= 000010      | CX.RUI= 000040  | D\$MNOD 000041   | FS.INI= 000000  | LA.WND= 004000   |
| CF.SYN= 000040      | CX.SMC= 000010  | D\$NA 000062     | FS.KIL= 000000  | LD\$LP = 000000  |
| CF.TRN= 000100      | CX.UNL= 000004  | D\$NBEA 000056   | FS.LCL= 100000  | LF.DRD= 000004   |
| CL\$MFL= 000010     | C\$SORE= 000400 | D\$NBRA 000054   | FS.LTM= 001000  | LF.FRC= 000001   |
| CL\$SFL= 000004     | C\$SRSH= 177564 | D\$NEND= 000054  | FS.MNT= 004000  | LF.HFO= 000010   |
| CL\$TYP= 000001     | C.ADD 000034    | D\$NLN 000030    | FS.MSN= 014000  | LF.HMF= 000040   |
| CL.MU1= 000001      | C.BID 000003    | D\$NN 000060     | FS.REA= 001000  | LF.HSF= 000020   |
| CL.MU2= 000002      | C.BUF 000014    | D\$OUTT 000043   | FS.RET= 000000  | LF.IRD= 000002   |
| CL.RES= 177774      | C.BUF1 000014   | D\$RETF 000050   | FS.REZ= 003000  | LF.MMF= 000200   |
| CMPDV = ***** GX    | C.BUF2 000024   | D\$RNN 000002    | FS.RLB= 002000  | LF.MSF= 000100   |
| CM.CIR= 000002      | C.CNT 000020    | D\$RTMR 000076   | FS.RNG= 011000  | LS.DLS= 100000   |
| CM.CON= 000200      | C.CNT1 000020   | D\$SEG 000036    | FS.RST= 000000  | LS.FCC= 000004   |
| CM.FMT= 100000      | C.CNT2 000030   | D\$SER 000032    | FS.RTN= 001000  | LS.FCO= 000001   |
| CM.HRD= 000002      | C.FLG 000022    | D\$SOLR 000052   | FS.SET= 005000  | LS.FCT= 000002   |
| CM.LIN= 000000      | C.FLG1 000022   | D\$SBUG= 177514  | FS.SFC= 005000  | LS.ILS= 100000   |
| CM.LOO= 000001      | C.FLG2 000032   | D\$SISK= 000000  | FS.SFR= 006000  | LS.MAK= 000020   |
| CM.XLO= 000004      | C.FNC 000010    | D\$SL11= 000001  | FS.SFS= 004000  | LS.MNK= 000040   |
| CP.DCF= 000040      | C.LIN 000006    | D\$SYNC= 000000  | FS.SPW= 040000  | LS.RES= 000360   |
| CP.HDL= 000007      | C.LNK 000000    | D\$SYNM= 000000  | FS.STM= 000000  | LS.RSV= 000300   |
| CP.PS = 177400      | C.MOD 000011    | ER\$ABM= 000010  | FS.STP= 002000  | LT.CCA= 000020   |
| CP.PSI= 000200      | C.NSP 000004    | ER\$ABO= 000046  | FS.STR= 001000  | LT.DIR= 000010   |
| CP.XCF= 000100      | C.PRO 000042    | ER\$ABT= 000011  | FS.TRM= 003000  | LT.LCL= 000001   |

```

178 .SBTTL Deallocate session control databases
179 ;+
180 ;**~DEALDB~Deallocate session control databases
181 ;
182 ; Deallocate those session control databases which have been allocated.
183 ;~
184 ; Inputs:
185 ; R5 = Address of database descriptor
186 ;
187 ; Registers modified:
188 ; R0, R1, R2, R3, R4
189 ;
190 000362 016500 000040 DEALDB: MOV N$LC+2(R5),R0 ; Get address of LLT table
191 000366 001410 BEQ 10$; If EQ, none allocated yet
192 000370 005065 000040 CLR N$LC+2(R5) ; Clear out allocation
193 000374 016501 000036 MOV N$LC(R5),R1 ; Get max # of logical links allowed
194 000400 005201 INC R1 ; Compute # of bytes to deallocate
195 000402 006301 ASL R1
196 000404 CALL @DEACB ; Deallocate the block
197
198 000410 10$: RETURN
199
200 000001 .END

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

```
.TITLE SESMN - Session control main logic
.IDENT /V05.00/
.ENABL LC

; Copyright (C) 1982, 1983, 1985 by
; Digital Equipment Corporation, Maynard, MASS.

; This software is furnished under a license for use only on a
; single computer system and may be copied only with the
; inclusion of the above copyright notice. This software, or
; any other copies thereof, may not be provided or otherwise
; made available to any other person except for use on such
; system and to one who agrees to these license terms. Title
; to and ownership of the software shall at all times remain
; in DEC.

; The information in this document is subject to change without
; notice and should not be construed as a commitment by Digital
; Equipment Corporation.

; DEC assumes no responsibility for the use or reliability of
; its software on equipment which is not supplied by DEC.

; Module description
;
; Session control main logic
;
; Ident history:
;
; 4.00 07-NOV-83
; DECNET-11M V4.0
; DECNET-11M-PLUS V2.0
;
; 5.00 22-JUL-85
; DECnet-11M/S V4.2
; DECnet-11M-Plus V3.0
; DECnet-Micro/RSX V1.0
;
```

SESMN11S CREATED BY MACRO ON 28-JUN-85 AT 19:57 PAGE 5 6 4  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE       | REFERENCES                                         |
|---------|-------------|----------------------------------------------------|
| TKTCB   | = ***** GX  | 8-123                                              |
| US.MDM  | = ***** GX  | 8-109                                              |
| US.MNT  | = ***** GX  | 8-110                                              |
| US.OFL  | = ***** GX  | 8-111                                              |
| U.ACP   | = ***** GX  | *8-112                                             |
| U.STS   | = ***** GX  | *8-109                                             |
| U.ST2   | = ***** GX  | *8-111                                             |
| WORD1   | = 000200    | #8-118 8-118 #8-121 8-121                          |
| WORD2   | = 000000    | #8-118 8-118 #8-121 8-121                          |
| \$CALLX | = ***** GX  | 8-103 8-114                                        |
| \$DSW   | = ***** GX  | 9-144                                              |
| \$SESDB | = ***** GX  | 8-92                                               |
| \$SESON | = 000000 RG | #8-76 10-164                                       |
| \$UCB   | = ***** GX  | 8-108                                              |
| \$XPTDC | = ***** GX  | 8-114                                              |
| \$XPTDS | = ***** GX  | 8-103                                              |
| \$XQTD  | = 000240 RG | 8-89 8-90 #9-141 9-147                             |
| \$GLB   | = *****     | 10-155 10-156                                      |
| \$T1    | = 000006    | #10-155 10-155 10-155 #10-156 10-156 10-156 10-156 |

6.5

```

351 .SBTTL Process received DI message
352
353 ;+
354 ***RCVDI-Process received DI message
355 This routine processes received DI messages.
356
357 Inputs:
358 R2 = Pointer to field following source link address
359 R4 = Address of CCB
360 R5 = Address of database descriptor
361
362 001022 112267 000000G RCVDI: MOVB (R2)+,$REASN ; Get reason code from message
363 001026 112267 000001G MOVB (R2)+,$REASN+1 ; ...
364
365 001032 CALL GETOPT ; Get optional data from message
366 001036 103450 BCS 100$; If CS, format error
367
368 001040 CALLE FNDLLT ; Find LLT associated with this link
369 001050 103443 BCS 100$; If CS, not match ... toss the message
370
371 001052 CALL SAVOPT ; Save optional data in LLT
372
373 001056 016763 000000G 000100 MOV $REASN,L,DCR(R3) ; Save reason for disconnect
374 001064 012765 000052 000022 MOV #ERCDI,NERRC(R5)
375 001072 121327 000006 CMPB (R3),#ST$CIR ; Does the user have the connect?
376 001076 001430 BEQ 100$; If EQ, yes ... toss message for now
377
378 .IF DF N$NCT
379 001100 016767 000000G 000000G MOV $OPLNG,$BYTE ; Count bytes received for DI message
380 001106 COUNT$ E$NBR ; ...
381 001114 COUNT$ E$NMR ; Count message received (DI)
382 001122 COUNT$ E$NMS ; Count message sent (DC)
383 .ENDC
384
385 001130 012702 001160' MOV #DISTA-2,R2 ; Point to substate table
386 001134 CALL KILLNK ; Kill the logical link
387
388 001140 016704 000000G MOV $RCCB,R4 ; Re-use DI buffer
389 001144 CALLE SENDDC ; Send DC response
390 001154 005067 000000G CLR $RCCB ; No CCB available now
391
392 001160 100$: RETURN
393
394 ;+
395 ; Received DI user and network substate table
396 ;+
397
398 001162 002 000 DISTA: .BYTE USCNF,NSDON ; CI sent
399 001164 012 000 .BYTE USEAC,NSDON ; Received CI, sent CC
400 001166 377 377 .BYTE -1,-1 ; Received CI
401 001170 006 000 .BYTE USDIS,NSDON ; Normal data transfer
402 001172 377 377 .BYTE -1,-1 ; Disconnect in progress
403 001174 004 000 .BYTE USDSC,NSDON ; Disconnect pending

```

H.5

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES | 11-343  | 11-347  | 12-368 | 12-389 | 13-419 | 16-557 | 16-569 | 18-621 | 18-624 |
|---------|------------|------------|---------|---------|--------|--------|--------|--------|--------|--------|--------|
| PROCON  | 001352 R   | 11-343     | 11-347  | 12-368  | 12-389 | 13-419 | 16-557 | 16-569 | 18-621 | 18-624 |        |
| RCPTBL  | 000004 R   | 9-200      | 11-318  | #15-498 |        |        |        |        |        |        |        |
| RCVCC   | 000552 R   | 7-70       | #7-76   |         |        |        |        |        |        |        |        |
| RCVCI   | 000120 R   | 8-129      | #11-286 |         |        |        |        |        |        |        |        |
| RCVCTL  | 000000 R   | 8-128      | 8-133   | #9-149  |        |        |        |        |        |        |        |
| RCVDC   | 001176 R   | 7-76       | #8-100  |         |        |        |        |        |        |        |        |
| RCVDI   | 001022 R   | 8-131      | #13-416 |         |        |        |        |        |        |        |        |
| RCVRCI  | 000460 R   | 8-130      | #12-362 |         |        |        |        |        |        |        |        |
| RCVTBL  | 000100 R   | 9-152      | #10-254 |         |        |        |        |        |        |        |        |
| REMLNK  | = ***** GX | 8-114      | #8-127  |         |        |        |        |        |        |        |        |
| RF.RTS  | = 000020   | 9-233      |         |         |        |        |        |        |        |        |        |
| RLSCI   | = ***** GX | 9-151      |         |         |        |        |        |        |        |        |        |
| RSNDDI  | 001654 RG  | 9-222      |         |         |        |        |        |        |        |        |        |
| RTRANS  | 001672 RG  | #17-593    |         |         |        |        |        |        |        |        |        |
| SAVOPT  | = ***** GX | 17-594     | #18-612 |         |        |        |        |        |        |        |        |
| SENDDC  | = ***** GX | 12-371     |         |         |        |        |        |        |        |        |        |
| SENDI   | 001552 RG  | 9-237      | 12-389  |         |        |        |        |        |        |        |        |
| SNSSED  | = ***** GX | 9-229      | #16-557 |         |        |        |        |        |        |        |        |
| STOPCC  | = ***** GX | 16-569     | 18-624  |         |        |        |        |        |        |        |        |
| STOPCI  | = ***** GX | 16-557     |         |         |        |        |        |        |        |        |        |
| ST\$CIR | = 000006   | 11-320     |         |         |        |        |        |        |        |        |        |
| ST\$CIS | = 000002   | 9-204      | 12-375  | 13-431  |        |        |        |        |        |        |        |
| ST\$DAT | = 000010   | 10-267     | 11-305  |         |        |        |        |        |        |        |        |
| ST\$DIP | = 000012   | 11-333     | 11-339  |         |        |        |        |        |        |        |        |
| T\$FLAG | 000044     | 9-227      |         |         |        |        |        |        |        |        |        |
| T\$LIF  | 000013     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LIFL | 000013     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LIFO | 000013     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LIFS | 000013     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LIN  | 000000     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LIPS | 000006     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LLD  | 000012     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LLDC | 000045     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LLDL | 000012     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LLDO | 000012     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LLDS | 000012     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LLEN | 000046     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LOPR | 000002     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LTCL | 000024     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LTIM | 000026     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LTPI | 000014     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$LTPS | 000020     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$NAPL | 000004     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$NFE  | 000000     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$NLEN | 000010     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$NNUL | 000002     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$NOPL | 000006     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$NRNI | 000042     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$NRPL | 000005     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$NRUL | 000007     | #6-51      |         |         |        |        |        |        |        |        |        |
| T\$NVR  | 000001     | #6-51      |         |         |        |        |        |        |        |        |        |

```

269 .SBTTL Disconnect completion
270 ;+
271 ; **--DISCMP--Disconnect completion
272 ;
273 ; The user has seen a disconnect/abort notice or it has been flushed
274 ; from the user's mailbox.
275 ;
276 ; Inputs:
277 ; R4 = Address of CCB
278 ; R5 = Address of database descriptor
279 ;
280 ; Registers modified:
281 ; R0, R1, R2
282 ;
283 000520 DISCMP::IF DF R$$$MPL
284 ;
285 SAVMAP ; Save current mapping
286 ;
287 .ENDC
288 ;
289 000520 SAVRG <R3> ; Get a free register
290 000522 016403 000012 MOV C.STS(R4),R3 ; Recover address of window block
291 000526 CALL @CCBRT ; Release the CCB
292 000532 016303 000004 MOV W.LLT(R3),R3 ; Get address of LLT
293 000536 CALLL ACCLLT ; Gain access to the LLT
294 ;
295 000546 112763 000000 000036 MOVB #US$DON,L.USTA(R3)
296 000554 016746 000000G MOV $IOPKT,-(SP) ; Save any I/O packet
297 000560 CALL RMVLNK ; Release the window block resources
298 000564 CALL IOSUC ; Complete any I/O
299 000570 012667 000000G MOV (SP)+,$IOPKT ; Restore any I/O packet
300 000574 RESRG <R3> ; Restore register
301 ;
302 .IF DF R$$$MPL
303 ;
304 RESMAP ; Restore mapping
305 ;
306 .ENDC
307 ;
308 000576 RETURN
309 ;
310 000001 .END

```

```

91 .SBTTL System level interface request - receive enable
92 ;+
93 ;*--.SERCE-System level interface request - receive enable
94 ;
95 ; This routine processes a single request for system level interface
96 ; service via the receive enable entry point.
97 ;
98 ; Inputs:
99 ; R3 = Subfunction code
100 ; R4 = Address of CCB
101 ; R5 = Address of database descriptor
102 ;
103 .SERCE::CALLR @RCETBL(R3) ; Dispatch to processing routine
104 ;
105 ;+
106 ; Receive enable dispatch table
107 ;
108 RCETBL: .WORD SLICRA ; Connect received acknowledgement
109 .WORD .+1 ; Return data segment
110 .WORD .+1 ; Return interrupt message
111 .WORD .+1 ; Disconnect received acknowledgement

```



```

588 .SBTTL Find node mapping by name
589
590 ;+
591 ;*-FNDNAM-Find node mapping by name
592 ; Scan the remote name table for a match with the specified node name.
593 ;-
594 ; Inputs:
595 ; R5 = Address of database descriptor
596 ; Outputs:
597 ; 'C' Clear - Valid node name mapping found
598 ; 'C' Set - Unable to find a node name mapping
599 ; Registers modified:
600 ; R0, R2, R3
601
602 FNDNAM: MOVB #LT.SLI,$LTYPE ; Indicate SLI style alias mapping
603 CALL MAPNAM ; Map node name to node address
604 BCS 100$; If CS, no mapping found
605 MOV N$SNOD(R5),$WORK; Fill in remote node address
606 100$: RETURN
607
608

```

SESSL111S CREATED BY MACRO ON 28-JUN-85 AT 19:59 PAGE 1 G 10  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE    | REFERENCES |
|---------|----------|------------|
| IS\$AS  | = *****  | 6-51       |
| NS\$EVL | = 000001 | #4-2       |
| NS\$SES | = 000001 | #6-57      |
| NS\$SLI | = *****  | 7-59       |
| RF.LOO  | = 100000 | #6-54      |
| RS\$11D | = *****  | 6-51       |
| RS\$11M | = 000000 | 6-51       |
| RS\$11S | = 000000 | 6-51       |
| R.ADD   | 000010   | #6-54      |
| R.FLAG  | 000012   | #6-54      |
| R.LEN   | 000014   | #6-54      |
| R.LNK   | 000000   | #6-54      |
| R.NAM   | 000002   | #6-54      |
| XS\$MCB | = *****  | 6-51       |
| ZF.COJ  | = 001000 | #6-51      |
| ZF.DDM  | = 000001 | #6-51      |
| ZF.DIA  | = 004000 | #6-51      |
| ZF.DLC  | = 000002 | #6-51      |
| ZF.DVP  | = 100000 | #6-51      |
| ZF.INI  | = 040000 | #6-51      |
| ZF.KMX  | = 000020 | #6-51      |
| ZF.LLC  | = 000004 | #6-51      |
| ZF.LMC  | = 000100 | #6-51      |
| ZF.MAN  | = 020000 | #6-51      |
| ZF.MFL  | = 000010 | #6-51      |
| ZF.MTM  | = 000400 | #6-51      |
| ZF.MUX  | = 000040 | #6-51      |
| ZF.PSE  | = 002000 | #6-51      |
| ZF.SLI  | = 010000 | #6-51      |
| ZF.TIM  | = 000200 | #6-51      |
| ZF.XSP  | = 000000 | #6-51      |
| ZS.ASN  | = 100000 | #6-51      |
| ZS.BSY  | = 140000 | #6-51      |
| Z.AVL   | 000014   | #6-51      |
| Z.DAT   | 000016   | #6-51      |
| Z.DSP   | 000000   | #6-51      |
| Z.FLG   | 000010   | #6-51      |
| Z.LEN   | = 000016 | #6-51      |
| Z.LLN   | 000006   | #6-51      |
| Z.MAP   | 000020   | #6-51      |
| Z.NAM   | 000004   | #6-51      |
| Z.PCB   | 000012   | #6-51      |
| Z.SCH   | 000007   | #6-51      |

```

352 .SBTTL Break the logical link
353 ;+
354 **--BRKLNK-Break the logical link
355 ;
356 This routine is called to destroy a logical in any state.
357 ;
358 Inputs:
359 R1 = Disconnect reason code
360 R3 = Virtual address of LLT
361 R5 = Address of database descriptor
362 ;
363 Registers modified:
364 R0, R1, R2, R4
365 ;
366 000744 010163 000100 BRKLNK: MOV R1,L.DCR(R3) ; Set up disconnect reason code
367 000750 012702 000756 MOV #DSTATE-2,R2 ; Point to disconnect substate table
368 000754 CALLR KILLNK ; Disconnect the link
369 ;
370 ;+
371 User and network substate table
372 ;
373 ;
374 000760 002 000 DSTATE: .BYTE USCNF,NSDON ; Connect initiate sent
375 000762 006 002 .BYTE USDIS,NSSDI ; Received CI, sent CC
376 000764 000 002 .BYTE USDON,NSSDI ; Received CI
377 000766 006 002 .BYTE USDIS,NSSDI ; Normal data transfer
378 000770 377 377 .BYTE -1,-1 ; Disconnect in progress
379 000772 004 002 .BYTE USDSC,NSSDI ; Disconnect pending

```

```

837 .SBTTL Flush I/O packets from a window block
838 ;+
839 ;**--FLSHIO--Flush I/O packets from a window block
840 ;
841 ; Flush any transmit or receive I/O packets from the window block.
842 ;
843 ; Inputs:
844 ; R4 = Address of window block
845 ; R5 = Address of database descriptor
846 ;
847 ; Registers modified:
848 ; R0, R1, R2, R3
849 ;
850 002100 FLSHIO::SAVMAP ; Save current mapping
851 002104 RECMAP ; Recover user apb mapping
852 002112 CALL FLSH1 ; Call routine in psect $HIGH
853 002116 RESMAP ; Restore previous mapping
854 002122 RETURN
855 ;
856 000222 .PSECT $HIGH
857 000222 FLSH1: SAVRG <R4> ; Save address of window block
858 000224 MOV W.SNDQ(R4),R3 ; Get pending transmit queue
859 000230 BEQ 10$; If EQ, no pending requests
860 000232 CLR W.SNDQ(R4) ; Clear out pending queue
861 000236 CALL FLSHRQ ; Flush all requests
862 000242 MOV (SP),R4 ; Recover window block address
863 ;
864 000244 10$: MOV W.RCVQ(R4),R3 ; Get pending receive queue
865 000250 BEQ 20$; If EQ, no pending requests
866 000252 CLR W.RCVQ(R4) ; Clear out pending queue
867 000256 CALL FLSHRQ ; Flush all requests
868 000262 MOV (SP),R4 ; Recover window block address
869 ;
870 000264 20$: MOV W.CINT(R4),R3 ; Get interrupt message transmit
871 000270 BEQ 30$; If EQ, no pending request
872 000272 CLR W.CINT(R4) ; Clear out pending queue
873 000276 CALL FLSHRQ ; Flush all requests
874 ;
875 000302 30$: RESRG <R4> ; Restore window block address
876 000304 RETURN

```

```

1302 .SBTTL Process descriptor name
1303 +
1304 **--PRONAM-Process descriptor name
1305
1306 Process the source or destination descriptor from the connect
1307 initiate message.
1308
1309 Inputs:
1310 R0 = Pointer into connect pending block
1311 R2 = Pointer to start of name descriptor in message
1312
1313 Outputs:
1314 R2 = Moved past name descriptor
1315 'C' Clear - Valid name descriptor processed
1316 'C' Set - Format error in name descriptor
1317
1318 Registers modified:
1319 R1, R5
1320
1321 PRONAM: MOVB (R2)+,R5 ; Get the format type code
1322 MOVB R5,(R0)+ ; and into pending connect block
1323 MOVB (R2)+,(R0)+ ; Store object type in pending connect block
1324 MOV #16,,R1 ; Set max field length
1325
1326 CMP #2,R5 ; Valid format type?
1327 BLO 30$; If LO, no ... C-bit set
1328 DEC R5 ; Format type 0?
1329 BMI 30$; If MI, yes ... all done
1330 BEQ 10$; If EQ, type 1 ... just move the data
1331
1332 MOVB (R2)+,(R0)+ ; Copy group code
1333 MOVB (R2)+,(R0)+ ; ...
1334 MOVB (R2)+,(R0)+ ; Copy user code
1335 MOVB (R2)+,(R0)+ ; ...
1336 SUB #4,R1 ; Reduce maximum field size
1337
1338 10$: SAVRG <R0,R1> ; Save registers
1339 TST (R0)+ ; Skip count field
1340 20$: MOVB #', (R0)+ ; Fill field with spaces
1341 SOB R1,20$; ...
1342 RESRG <R1,R0> ; Restore registers
1343
1344 30$: CALL PROIMG ; Process image field
1345 RETURN

```

|                 |                  |                      |                     |                 |
|-----------------|------------------|----------------------|---------------------|-----------------|
| EV\$HFE= 000506 | E\$NLLA 000012   | FR\$CDF= 000002      | IN.ILS= 000001      | L.ILSQ 000052   |
| EV\$IFL= 000413 | E\$NLNK 000000   | FR\$DAO= 000011      | IOFIN = ***** GX    | L.ILTT 000066   |
| EV\$IFD= 000415 | E\$NML 000040    | FR\$EXC= 000000      | 002 IDRED0 000452RG | L.LDA 000032    |
| EV\$IFS= 000414 | E\$NMNR 000024   | FR\$FRM= 000010      | I\$RAR= 000000      | L.LIA 000034    |
| EV\$INF= 000515 | E\$NMS 000030    | FR\$FTL= 000005      | I\$RDN= 000000      | L.LLA 000002    |
| EV\$LDL= 000407 | E\$NNOD 000002   | FR\$OPN= 000004      | I.PRM = ***** GX    | L.LNG 000124    |
| EV\$LDN= 010416 | E\$NOD 000010    | FR\$RFD= 000006      | I.TCB = ***** GX    | L.LNO 000026    |
| EV\$LDO= 000411 | E\$NRT 000042    | FR\$SBD= 000012      | KILLNK 002364RG     | L.LPT 000065    |
| EV\$LDS= 000410 | E\$NRTP 000005   | FR\$SHO= 000003      | KISAR6= ***** GX    | L.LSA 000030    |
| EV\$LSC= 000500 | E\$NSEG 000010   | FR\$UBU= 000013      | KSAR6 = ***** GX    | L.LSFD 000046   |
| EV\$LUP= 000412 | E\$NTIM 000046   | FR\$UPT= 000014      | K\$CNT= 177546      | L.LSFI 000044   |
| EV\$NOL= 000402 | E\$NUSE 000004   | FS.AST= 000000       | K\$CSR= 177546      | L.LTT 000062    |
| EV\$NRC= 000416 | E\$PORT 000014   | FS.CIB= 002000       | K\$SLDC= 000000     | L.MASQ 000070   |
| EV\$NSC= 000200 | E\$PRM 000002    | FS.CRA= 001000       | K\$TPS= 000074      | L.MAST 000073   |
| EV\$NUL= 000401 | E\$STAT 000006   | FS.DIS= 013000       | LA.ACK= 100000      | L.MASZ 000072   |
| EV\$NVR= 000406 | E\$STRT 000006   | FS.DVC= 001000       | LA.CRS= 020000      | L.NIN 000020    |
| EV\$OPL= 000403 | E\$TCB 000004    | FS.ENB= 012000       | LA.MSK= 170000      | L.NXN 000016    |
| EV\$PCC= 034000 | E\$XPR= 000000   | FS.EXI= 001000       | LA.NAK= 110000      | L.NXTH 000010   |
| EV\$PCI= 034001 | E.CTL 000020     | FS.GET= 006000       | LA.NMS= 010000      | L.DPD 000103    |
| EV\$PCM= 034002 | E.DATA 000046    | FS.HLT= 000000       | LA.RES= 040000      | L.OPDL 000102   |
| EV\$PFE= 000404 | E.EVT 000002     | FS.INI= 000000       | LA.WND= 004000      | L.REM 000006    |
| EV\$PFC= 034003 | E.LCN 000042     | FS.KIL= 000000       | LDBGT = ***** GX    | L.RFC 000050    |
| EV\$RCF= 000517 | E.LEN 000216     | FS.LCL= 100000       | LDBRT = ***** GX    | L.RLA 000004    |
| EV\$RDC= 010001 | E.LIN 000024     | FS.LTM= 001000       | LD\$LP = 000000     | L.RNO 000022    |
| EV\$RDR= 010002 | E.LNK 000000     | FS.MNT= 004000       | LF.DRD= 000004      | L.RTQ 000060    |
| EV\$RJE= 035106 | E.MOD 000036     | FS.MSN= 014000       | LF.FRC= 000001      | L.RTYD 000055   |
| EV\$RSC= 000501 | E.NOD 000034     | FS.REA= 001000       | LF.HFO= 000010      | L.RTYI 000057   |
| EV\$RUL= 000405 | E.PDV 000021     | FS.RET= 000000       | LF.HMF= 000040      | L.SEC 000064    |
| EV\$SNA= 035000 | E.PORT 000040    | FS.REZ= 003000       | LF.HSF= 000020      | L.SEGZ 000076   |
| EV\$SNF= 000516 | E.PRM 000026     | FS.RLB= 002000       | LF.IRD= 000002      | L.STA 000000    |
| EV\$SPE= 035001 | E.PVC 000044     | FS.RNG= 011000       | LF.MMF= 000200      | L.TC 000042     |
| EV\$XCE= 034110 | E.SIZ 000022     | FS.RST= 000000       | LF.MSF= 000100      | L.TIC 000043    |
| EV\$XDI= 013600 | E.TIME 000004    | FS.RTN= 001000       | LS.DLS= 100000      | L.TIPD 000013   |
| EV\$XGW= 034111 | FC.CCP= 000020   | FS.SET= 005000       | LS.FCC= 000004      | L.TIPI 000012   |
| EV\$XMX= 000514 | FC.CTL= 000006   | FS.SFC= 005000       | LS.FCO= 000001      | L.TMRD 000054   |
| EV\$XRS= 000512 | FC.KCP= 000016   | FS.SFR= 006000       | LS.FCI= 000002      | L.TMRI 000056   |
| EV\$XSC= 000513 | FC.KIL= 000004   | FS.SFS= 004000       | LS.ILS= 100000      | L.TYP 000001    |
| EV\$X2S= 013500 | FC.MAN= 000024   | FS.SPW= 040000       | LS.MAK= 000020      | L.USA 000024    |
| EV.CCB= 000001  | FC.MLD= 000026   | FS.STM= 000000       | LS.MNK= 000040      | L.USTA 000036   |
| EV.CIR= 000020  | FC.PCT= 000030   | FS.STP= 002000       | LS.RES= 000360      | L.VER 000015    |
| EV.LCB= 000100  | FC.PWR= 000022   | FS.STR= 001000       | LS.RSV= 000300      | L.WIND 000040   |
| EV.LIN= 000004  | FC.RCE= 000002   | FS.TRM= 003000       | LT.CCA= 000020      | MAPOBJ 002430RG |
| EV.MAP= 000002  | FC.RCP= 000014   | FS.WLB= 001000       | LT.DIR= 000010      | MA.CI = 000040  |
| EV.MOD= 000040  | FC.TIM= 000010   | FS.XKL= 002000       | LT.LCL= 000001      | MA.DA = 000000  |
| EV.NOD= 000010  | FC.XCP= 000012   | FS.XOF= 010000       | LT.LPL= 000002      | MA.IL = 000020  |
| EV.PRT= 000200  | FC.XME= 000000   | FS.XON= 007000       | LT.NOT= 000040      | MC.CC = 000040  |
| E\$DATA 000020  | FDISP 000340R    | FS.ZER= 002000       | LT.RSU= 000200      | MC.CI = 000020  |
| E\$EVTS 000000  | FLCON 000356R    | 002 F\$SLVL= 000001  | LT.SLI= 000C04      | MC.DC = 000100  |
| E\$LCN 000016   | FLSHIO 002100RG  | 002 GETLDB 002242RG  | LT.TDA= 000100      | MC.DI = 000060  |
| E\$LIN 000000   | FLSHMB 002124RG  | 002 GETSDB 002266RG  | L\$ASG= 000000      | MC.NO = 000000  |
| E\$MOD 000012   | FLSHRQ 000306R   | 002 G\$STPP= 000000  | L\$DRV= 000000      | MC.RC = 000140  |
| E\$NBR 000014   | FLSHI 000222R    | 002 G\$STSS= 000000  | L\$SP11= 000001     | MD.BM = 000040  |
| E\$NBS 000020   | FLSLST= ***** GX | 002 G\$STTK= 000000  | L\$11R= 000000      | MD.EM = 000100  |
| E\$NCR 000034   | FLVfy 000356R    | 002 G\$SWRD= 000000  | L.CSTA 000037       | MD.ILS= 000040  |
| E\$NCS 000036   | FNDMBX 000412RG  | 002 H.LUN = ***** GX | L.CTR 000074        | MD.IM = 000020  |
| E\$NIC 000044   | FR\$BCC= 000007  | IE.ABO = ***** GX    | L.DCR 000100        | MF.ACK= 000004  |
| E\$NLEN 000050  | FR\$CCF= 000001  | IN.DAT= 000400       | L.FLAG 000014       | MF.CTL= 000010  |

```

SSSSSSSS EEEEEEEEE SSSSSSSS TTTTTTTTT CCCCCCCC BBBB8888
SSSSSSSS EEEEEEEEE SSSSSSSS TTTTTTTTT CCCCCCCC BBBB8888
SS EE SS TT CC BB BB
SS EE SS TT CC BB BB
SS EE SS TT CC BB BB
SS EE SS TT CC BB BB
SSSSSS EEEEEEEEE SSSSSS TT CC BBBB8888
SSSSSS EEEEEEEEE SSSSSS TT CC BBBB8888
 SS SS TT CC BB BB
 SS SS TT CC BB BB
 SS SS TT CC BB BB
 SS SS TT CC BB BB
SSSSSSS EEEEEEEEE SSSSSSSS TT CCCCCCCC BBBB8888
SSSSSSS EEEEEEEEE SSSSSSSS TT CCCCCCCC BBBB8888

```

....  
....  
....  
....

```

11 11 SSSSSSSS
11 11 SSSSSSSS
1111 1111 SS
1111 1111 SS
11 11 SS
11 11 SS
11 11 SSSSSS
11 11 SSSSSS
11 11 SS
11 11 SS
11 11 SS
11 11 SS
111111 111111 SSSSSSSS
111111 111111 SSSSSSSS

```

6 16

```

109 .SBTTL Session control timer service action routines
110 ;+
111 ;**--TIMXXX-Session control timer service action routines
112 ;
113 ; These routines are called to time session control services for a
114 ; logical link.
115 ;
116 ; Inputs:
117 ; R2 = Address of data subchannel timer
118 ; R3 = Virtual address of LL7
119 ; R5 = Address of database descriptor
120 ;
121 ; Registers modified:
122 ; R0, R1, R2, R3, R4
123 ;
124 .ENABL LSB
125 ;+
126 ;**--TIMCIS-Connect initiate sent
127 ;**--TIMCC-Connect initiate received, connect confirm sent
128 ;
129 ;
130 000110 017701 0000006 TIMCIS: MOV @DECPT,R1 ; Get address of DEC home block
131 000114 116165 000043 000014 MOV D$OUTT(R1),NSDLY(R5) ; Reset infinity for outgoing connects
132 000122 105712 TIMCC: TSTB (R2) ; Is the timer running?
133 000124 001423 BEQ 20$; If EQ, no
134 000126 105322 DECB (R2)+ ; Reduce time to go
135 000130 001021 BNE 20$; If NE, timer still running
136 000132 105312 DECB (R2) ; Have we exhausted the retry count?
137 000134 003402 BLE 10$; If LE, yes ... break the logical link
138 000136 CALLR RTRANS ; Retransmit the CC or CI message
139 ;
140 000142 116301 000065 10$: MOV L,LPT(R3),R1 ; Copy possible 'object did not respond'
141 000146 001002 BNE 15$; If NE, that assumption is correct
142 000150 012701 000047 MOV #ER$COM,R1 ; Set up disconnect reason code
143 000154 CALLR BRKLNK ; Break the logical link
144 ;+
145 ;**--TIMCIR-Connect initiate received
146 ;
147 ;
148 000160 005713 TIMCIR: TST (R3) ; Did we have a resource allocation failure?
149 000162 100004 BPL 20$; If PL, no
150 000164 CALLE ACKCI ; Try to resend the CI ACK
151 000174 20$: RETURN
152 ;
153 ;+
154 ;**--TIMDIP-Disconnect in progress
155 ;
156 ;
157 000176 TIMDIP: CALL TRMUSR ; Process user disconnect state
158 000202 032713 000400 BIT #LCL*400,(R3) ; Is this a local logical link?
159 000206 001004 BNE 30$; If NE, yes ... don't worry about timers
160 ;
161 000210 105712 TSTB (R2) ; Is the isconnect timer running?
162 000212 001402 BEQ 30$; If EQ, yes
163 000214 105312 DECB (R2) ; Count it down the timer
164 000216 001366 BNE 20$; If NE, still active
165 ;

```

H 16



|                  |                 |                   |                  |                   |     |
|------------------|-----------------|-------------------|------------------|-------------------|-----|
| L.T.LPL= 000002  | L.TIPI 000012   | NFSMOU= 000040    | N.C9A 000142     | SCNGNQ 000034R    | 002 |
| L.T.NOT= 000040  | L.TMRD 000054   | NFSRST= 000002    | N.CDAC 000140    | SESTIM= ***** GX  |     |
| L.T.RSU= 000200  | L.TMRI 000056   | NFSSCN= 000020    | N.CDDS 000070    | SRSTD = ***** GX  |     |
| L.T.SLI= 000004  | L.TYP 000001    | NFSSHU= 000004    | N.CDEV 000062    | STPCT = ***** GX  |     |
| L.T.TDA= 000100  | L.USA 000024    | NFSTIM= 000200    | N.CID 000064     | STSCC = 000004    |     |
| L.S\$ASG= 000000 | L.USTA 000036   | NMCLH = ***** GX  | N.CIDC 000062    | STSCIR= 000006    |     |
| L.S\$DRV= 000000 | L.VER 000015    | NMCON 000370R     | N.CPS 000106     | STSCIS= 000002    |     |
| L.S\$P11= 000001 | L.WIND 000040   | NMEVT 000376R     | N.CPSC 000104    | STSDAT= 000010    |     |
| L.S\$11R= 000000 | MA.CI = 000040  | NMVFY 000406R     | N.CTL 000000     | STSDIP= 000012    |     |
| L.CSTA 000037    | MA.DA = 000000  | NS\$DCN= 000000   | N.CUIC 000066    | STSPND= 000014    |     |
| L.CTR 000074     | MA.IL = 000020  | NS\$SDI= 000002   | N.CUNI 000064    | S\$SWRG= 000000   |     |
| L.DCR 000100     | MC.CC = 000040  | NS\$WDC= 000004   | N.DDE 000010     | S\$SYSZ= 007600   |     |
| L.FLAG 000014    | MC.CI = 000020  | NSACQ 000000      | N.DDEC 000006    | TKTCB = ***** GX  |     |
| L.ILSQ 000052    | MC.DC = 000100  | NSACTL 000032     | N.DFM 000004     | TSKRT = ***** GX  |     |
| L.ILTT 000066    | MC.DI = 000060  | NSCIR 000034      | N.DGP 000006     | T\$KMG= 000000    |     |
| L.LDA 000032     | MC.NO = 000000  | NSDLA 000020      | N.DNM 000014     | T\$MIN= 000000    |     |
| L.LIA 000034     | MC.RC = 000140  | NSDLY 000014      | N.DNMC 000012    | T.PCB = ***** GX  |     |
| L.LLA 000002     | MD.BM = 000040  | NSLEN 000054      | N.DOT 000005     | T.RCYL= ***** GX  |     |
| L.LNG 000124     | MD.EM = 000100  | NSENC 000042      | N.DUS 000010     | T.ST3 = ***** GX  |     |
| L.LNO 000026     | MD.ILS= 000040  | NSERRC 000022     | N.SDE 000042     | T3.REM= ***** GX  |     |
| L.LPT 000065     | MD.IM = 000020  | NSFLG 000005      | N.SDEC 000040    | UISAR6= ***** GX  |     |
| L.LSA 000030     | MF.ACK= 000004  | NS\$NC 000006     | N.SEGZ 000002    | US\$CNF= 000002   |     |
| L.LSFD 000046    | MF.CIL= 000010  | NSGENQ 000052     | N.SFM 000036     | US\$DIS= 000006   |     |
| L.LSFI 000044    | MF.DAT= 000000  | NSGTM 000015      | N.SGP 000040     | US\$DON= 000000   |     |
| L.LTT 000062     | M\$SCRB= 000124 | NSHIGH 000033     | N.SND 000030     | US\$DSC= 000004   |     |
| L.MASQ 000070    | M\$SCRX= 000000 | NSLLT 000026      | N.SNM 000046     | US\$EAC= 000012   |     |
| L.MAST 000073    | M\$SFCS= 000000 | NSLLTM 000024     | N.SNMC 000044    | US\$WDS= 000010   |     |
| L.MASZ 000072    | M\$SMGE= 000000 | NSLVC 000036      | N.SOI 000037     | VFYNAM= ***** GX  |     |
| L.NIN 000020     | M\$SMUP= 000000 | NSMBXQ 000050     | N.SUS 000042     | V\$SCTR= 001000   |     |
| L.NXN 000016     | M\$SNET= 000000 | NSPLT 000030      | P\$P45= 000000   | X\$SDBT= 000000   |     |
| L.NXTH 000010    | M\$SOVR= 000000 | NSSLA 000016      | P\$WRD= 000000   | XCLQIO= ***** GX  |     |
| L.OPD 000103     | M.MAIL 000014   | NSSNOD 000012     | P.HDR = ***** GX | SCNQIO= ***** GX  |     |
| L.OPDL 000102    | M.MAX 000011    | NSTIM 000004      | QIDOSP 000026R   | SCOPT = ***** GX  |     |
| L.REM 000006     | M.MBL 000020    | NSVCB 000010      | QRMVF = ***** GX | SCTQIO= ***** GX  |     |
| L.RFC 000050     | M.NAST 000007   | NS\$ACC= 000001   | Q\$SOPT= 000010  | SDMQIO= ***** GX  |     |
| L.RLA 000004     | M.NEXT 000002   | NS\$EVL= 000001   | RDBRT = ***** GX | SDSQIO= ***** GX  |     |
| L.RNO 000022     | M.RESP 000016   | NS\$LDV= 000001   | REJECT= ***** GX | \$IOPKT= ***** GX |     |
| L.RTO 000060     | M.SPA 000012    | NS\$MLL= 000001   | REQTSK 000316R   | \$RCCB = ***** GX |     |
| L.RTYD 000055    | M.TASK 000004   | NS\$MOV= 000010   | REQTS1 000334R   | \$ROCPY= ***** GX |     |
| L.RTYI 000057    | M.USE 000010    | NS\$NCT= 000001   | RET 000350R      | \$ROTCB= ***** GX |     |
| L.SEC 000064     | NC.FM0= 000000  | NS\$PEM= 000001   | R\$SDER= 000000  | \$SESPD= ***** GX |     |
| L.SEGZ 000076    | NC.FM1= 000001  | NS\$SES= 000001   | R\$SK11= 000001  | \$SSHFT= 000001   |     |
| L.STA 000000     | NC.FM2= 000002  | NS\$SMC= ***** GX | R\$SND= 000000   | \$S\$ = 000062    |     |
| L.TC 000042      | NDISP 000352R   | N.CAC 000120      | R\$S11M= 000000  | .SERCP= ***** GX  |     |
| L.TIC 000043     | NF\$BLK= 000100 | N.CACC 000116     | R\$S11S= 000000  | .S\$S\$ = 000034  |     |
| L.TIPD 000013    | NF\$DMO= 000010 | N.CBL = 000142    |                  |                   |     |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000516 001 (RW,I,LCL,REL,CON)  
\$HIGH 000414 002 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 74  
Work file writes: 80

|                  |                  |                 |                 |                 |
|------------------|------------------|-----------------|-----------------|-----------------|
| AC\$DNT= 000002  | DSIPL 000051     | EV\$NOL= 000402 | ESNUSE 000004   | MO\$29S= 000010 |
| AC\$X25= 000001  | DSLID 000020     | EV\$NRC= 000416 | ESPDRT 000014   | MSHIGH= 000003  |
| AE\$CIR= 000003  | DSLNAM 000006    | EV\$NSC= 000200 | ESPRM 000002    | MS\$CRB= 000124 |
| AE\$LIN= 000001  | DSLNUM 000014    | EV\$NUL= 000401 | ESTAT 000006    | MS\$CRX= 000000 |
| AE\$MOD= 000004  | DSLST 000047     | EV\$NVR= 000406 | ESTRT 000006    | MS\$FCS= 000000 |
| ALDCB = ***** GX | DSMAXC 000064    | EV\$DPL= 000403 | ESTCB 000004    | MS\$MGE= 000000 |
| ALDCDB 000274R   | DSMAXH 000066    | EV\$PCC= 034000 | ES\$XPR= 000000 | MS\$MUP= 000000 |
| AS\$CHK= 000000  | DSMAXV 000070    | EV\$PCI= 034001 | E.CTL 000020    | MS\$NET= 000000 |
| AS\$CPS= 000000  | DSMLL 000040     | EV\$PCM= 034002 | E.DATA 000046   | MS\$DVR= 000000 |
| AS\$PRI= 000000  | DSMDD 000041     | EV\$PFE= 000404 | E.EVT 000002    | MS\$300= 000000 |
| AS\$TRP= 000000  | DSNA 000062      | EV\$PPC= 034003 | E.LCN 000042    | MS\$301= 000001 |
| BYTE3 = 000300   | DSNBEA 000056    | EV\$RCF= 000517 | E.LEN 000216    | MS\$302= 000002 |
| CL\$ASZ= 010500  | DSNBRA 000054    | EV\$RDC= 010001 | E.LIN 000024    | MS\$303= 000003 |
| CL\$DLL= 000500  | DSNEND= 000054   | EV\$RDR= 010002 | E.LNK 000000    | NF\$BLK= 000100 |
| CL\$ECL= 000300  | DSNLN 000030     | EV\$RJE= 035106 | E.MDD 000036    | NF\$DMD= 000010 |
| CL\$LDN= 010400  | DSNN 000060      | EV\$RSC= 000501 | E.NDD 000034    | NF\$MOU= 000040 |
| CL\$MAN= 000000  | DSOUT 000043     | EV\$RUL= 000405 | E.PDV 000021    | NF\$RST= 000002 |
| CL\$PAZ= 034100  | DSRETF 000050    | EV\$SNA= 035000 | E.PDRT 000040   | NF\$SCN= 000020 |
| CL\$PLH= 034000  | DSRNN 000002     | EV\$SNF= 000516 | E.PRM 000026    | NF\$SHU= 000004 |
| CL\$PLL= 000600  | DSRTMR 000076    | EV\$SPE= 035001 | E.PVC 000044    | NF\$TIM= 000200 |
| CL\$PR1= 034200  | DSSEG 000036     | EV\$XCE= 034110 | E.SIZ 000022    | NM\$ARA= 176000 |
| CL\$ROU= 010000  | DSSEF 000032     | EV\$XDI= 013600 | E.TIME 000004   | NM\$NDD= 001777 |
| CL\$SES= 000200  | DS\$QRL 000052   | EV\$XGW= 034111 | FR\$BCC= 000007 | NT\$AKD= 000020 |
| CL\$SGE= 035000  | DS\$BUG= 177514  | EV\$XMX= 000514 | FR\$CCF= 000001 | NT\$AKI= 000022 |
| CL\$SSE= 035100  | DS\$ISK= 000000  | EV\$XRS= 000512 | FR\$CDF= 000002 | NT\$CC = 000016 |
| CL\$TRN= 000400  | DS\$11= 000001   | EV\$XSC= 000513 | FR\$DAD= 000011 | NT\$CDN= 000000 |
| CL\$XL2= 013700  | DS\$YNC= 000000  | EV\$X2S= 013500 | FR\$EXC= 000000 | NT\$CTL= 000000 |
| CL\$XL3= 013600  | DS\$YNM= 000000  | EV.CCB= 000001  | FR\$FRM= 000010 | NT\$DAT= 000002 |
| CL\$X2S= 013500  | D.NAM = ***** GX | EV.CIR= 000020  | FR\$FTL= 000005 | NT\$DC = 000012 |
| C\$DRE= 000400   | D.UCB = ***** GX | EV.LCB= 000100  | FR\$DPN= 000004 | NT\$DIS= 000014 |
| C\$SRSH= 177564  | D.UNIT= ***** GX | EV.LIN= 000004  | FR\$RFD= 000006 | NT\$DLS= 000006 |
| DEACB = ***** GX | EF\$ACT= 000001  | EV.MAP= 000002  | FR\$SBU= 000012 | NT\$ILS= 000010 |
| DEALDB 000362RG  | EVLSES= ***** GX | EV.MDD= 000040  | FR\$SHO= 000003 | NT\$IMS= 000002 |
| DECP1 = ***** GX | EV\$ACF= 000201  | EV.NDD= 000010  | FR\$SUB= 000013 | NT\$INT= 000004 |
| DEVHD = ***** GX | EV\$ADR= 000420  | EV.PR1= 000200  | FR\$UPT= 000014 | NT\$RET= 000032 |
| DL\$AST= 000002  | EV\$ADU= 000417  | ESDATA 000020   | FS\$LVL= 000001 | NT\$ROU= 000024 |
| DL\$HLT= 000000  | EV\$APL= 000400  | ES\$VTS 000000  | GS\$TTP= 000000 | NT\$RTR= 000030 |
| DL\$IST= 000001  | EV\$ARC= 000421  | ESLCN 000016    | GS\$TSS= 000000 | NT\$TSP= 000026 |
| DL\$MAI= 000004  | EV\$AUC= 003010  | ESLIN 000000    | GS\$TTK= 000000 | NS\$ACQ 000000  |
| DL\$DFF= 000001  | EV\$AUS= 000003  | ESMOD 000012    | GS\$WRD= 000000 | NS\$ACTL 000032 |
| DL\$ON = 000000  | EV\$CDF= 000520  | ESNBR 000014    | IS\$RAR= 000000 | NSCIR 000034    |
| DL\$RJN= 000003  | EV\$CDZ= 000011  | ESNBS 000020    | IS\$RDN= 000000 | NSDLA 000020    |
| DL\$SHU= 000002  | EV\$DBR= 000302  | ESNCR 000034    | K\$SCNT= 177546 | NSDLY 000014    |
| DL\$SYN= 000005  | EV\$GAS= 035101  | ESNCS 000036    | K\$SCSR= 177546 | NSELEN 000054   |
| DS\$AMX 000072   | EV\$HCE= 035114  | ESNIC 000044    | K\$SLDC= 000000 | NSENC 000042    |
| DS\$AMXC 000074  | EV\$HCI= 035113  | ESNLEN 000050   | K\$STPS= 000074 | NSERRC 000022   |
| DS\$ANN 000000   | EV\$HFE= 000506  | ESNLLA 000012   | LD\$LP = 000000 | NSFLG 000005    |
| DS\$BRP 000102   | EV\$IFL= 000413  | ESNLNK 000000   | L\$SASG= 000000 | NSFNC 000006    |
| DS\$BRT 000100   | EV\$IFD= 000415  | ESNML 000040    | L\$SDRV= 000000 | NSGENQ 000052   |
| DS\$ELF 000045   | EV\$IFS= 000414  | ESNMR 000024    | L\$SP11= 000001 | NSGTM 000015    |
| DS\$ELW 000046   | EV\$INF= 000515  | ESNMS 000030    | L\$111R= 000000 | NSHIGH 000033   |
| DS\$END = 000104 | EV\$LDL= 000407  | ESNNDD 000002   | MO\$SAC= 000016 | NSLLT 000026    |
| DS\$FNB 000034   | EV\$LDN= 010416  | ESNOD 000010    | MO\$SPR= 000012 | NSLLTM 000024   |
| DS\$H1OR 000024  | EV\$LDN= 000411  | ESNRT 000042    | MO\$SSV= 000014 | NSLVC 000036    |
| DS\$HOST 000022  | EV\$LDS= 000410  | ESNRTF 000005   | MO\$25A= 000006 | NSMBXQ 000050   |
| DS\$INAC 000044  | EV\$LSC= 000500  | ESNSEG 000010   | MO\$25P= 000002 | NSPLLT 000030   |
| DS\$INCT 000042  | EV\$LUP= 000412  | ESNTIM 000046   | MO\$25S= 000004 | NSSLA 000016    |

```

42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,CALLX,EVT$,MAPLLT,RET
45 .MCALL RQST$,DIR$,WSIG$
46 .MCALL EVLDF$,ECDDB$
47
48 .IF DF R$$MPL
49 .IF NDF R$$PRO
50 .MCALL GIN$
51 .ENDC
52 .ENDC
53
54 EVLDF$; Define event logger offsets
55 ECDDB$; Define ECL database offsets
56
57 000000 N$$SES = 1 ; This module is part of session control
58 000001

```

SESMN11S CREATED BY MACRO ON 28-JUN-85 AT 19:57 PAGE 6 H 4  
 MACRO CROSS REFERENCE CREF 04.00

| MACRO NAME | REFERENCES                    |
|------------|-------------------------------|
| CALL       | 8-83 8-85 8-93 8-102 8-113    |
| CALLR      | 8-124                         |
| CALLX      | #6-44 8-103 8-114             |
| DIR\$      | #6-45 9-141 #9-146 9-146      |
| ECDDBS     | #6-46 6-55                    |
| EVLDF\$    | #6-46 6-54                    |
| EVT\$      | #6-44 8-118 8-121             |
| MAP        | #6-44                         |
| MAPLLT     | #6-44                         |
| OFF\$      | #10-155 #10-156               |
| RESRG      | #6-44                         |
| RETC       | #6-44                         |
| RETURN     | 8-87 8-106                    |
| RQST\$     | #6-45 10-155 10-156           |
| R50\$      | #10-155 10-155 #10-156 10-156 |
| SAVRG      | #6-44                         |
| SWSTK\$    | 8-83 8-102                    |
| WSIG\$S    | #6-45 9-146                   |
| XQT        | #7-61 8-89 8-90               |

```

405 .SBTTL Process received DC message
406 ;+
407 ;**--RCVDC-Process received DC message
408 ;
409 ; This routine processes received DC messages.
410 ;
411 ; Inputs:
412 ; R2 = Pointer to field following source link address
413 ; R4 = Address of CCB
414 ; R5 = Address of database descriptor
415
416 001176 112267 000000G RCVDC: MOVB (R2)+,$REASN ; Get reason code from message
417 001202 112267 000001G MOVB (R2)+,$REASN+1 ; ...
418
419 001206 CALLE FNDLLT ; Find LLT associated with the link
420 001216 103431 BCS 100$; If CS, no match ... toss the message
421
422 001220 COUNT$ E$NMR ; Count message received
423 001226 026563 000016 000004 CMP N$SLA(R5),L.RLA(R3)
424 001234 001022 BNE 100$; If NE, not for this logical link
425 001236 016700 MOV $REASN,R0 ; Get the reason code
426 001242 020027 CMP R0,ERSTA ; Is this a 'NO LINK' message?
427 001246 001002 BNE 10$; If NE, no
428 001250 012700 MOV #ER$ABD,R0 ; Change code to session control error
429
430 001254 010063 000100 10$: MOV R0,L.DCR(R3) ; Save disconnect reason code
431 001260 121327 000006 CMPB (R3),#ST$CIR ; Does user have the connect?
432 001264 001406 BEQ 100$; If EQ, yes ... toss message for now
433 001266 012702 001302' MOV #DCSTA-2,R2 ; Point to disconnect substate table
434 001272 005063 000054 CLR L.TMRD(R3) ; No need to send any more DI's
435 001276 CALL KILLNK ; Kill the logical link
436
437 001302 100$: RETURN
438
439 ;+
440 ; Received DC user and network substate table
441 ;
442
443 001304 002 000 DCSTA: .BYTE USCNF,NSDON ; CI sent
444 001306 012 000 .BYTE USEAC,NSDON ; Received CI, sent CC
445 001310 377 377 .BYTE -1,-1 ; Received CI
446 001312 006 000 .BYTE USDTC,NSDON ; Normal data transfer
447 001314 377 000 .BYTE -1,NS$DON ; Disconnect in progress
448 001316 004 000 .BYTE USDSC,NSDON ; Disconnect pending

```

SESPRO11S CREATED BY MACRO ON 28-JUN-85 AT 19:57 PAGE 4 H 6  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                                                   |
|---------|------------|--------------------------------------------------------------|
| T\$RPRI | 000040     | #6-51                                                        |
| T\$SVC  | 000034     | #6-51                                                        |
| T\$T5   | 000030     | #6-51                                                        |
| T\$T6   | 000032     | #6-51                                                        |
| USRCC   | = ***** GX | 11-331                                                       |
| USRCI   | = ***** GX | 9-206                                                        |
| USRINT  | = ***** GX | 7-77                                                         |
| US\$CNF | = 000002   | 12-398 13-443                                                |
| US\$DIS | = 000006   | 12-401 13-446                                                |
| US\$DSC | = 000004   | 12-403 13-448                                                |
| US\$EAC | = 000012   | 12-399 13-444                                                |
| \$BYTE  | = ***** GX | *11-309 *12-379                                              |
| \$CALLX | = ***** GX | 8-118 9-237 10-264 11-302 11-316 11-320 11-343 11-347        |
|         |            | 12-368 12-389 13-419 16-557 16-369 18-624                    |
| \$INFO  | = ***** GX | *9-158 *11-287 15-511                                        |
| \$OPDAT | = ***** GX | 14-476                                                       |
| \$OPLNG | = ***** GX | 11-309 12-379 *14-474                                        |
| \$RCCB  | = ***** GX | 8-116 9-226 *9-230                                           |
| \$REASN | = ***** GX | *12-362 *12-363 12-373 *13-416 *9-238 *11-348 12-388 *12-390 |
| \$SEGMT | = ***** GX | *9-161 *9-162 *11-290 *11-291 13-425 *16-558 16-564 16-565   |
| \$SRVCS | = ***** GX | *9-157 *9-164 *11-286 11-293 15-520                          |
| .SERCP  | 000000 RG  | #7-70                                                        |

|                  |                 |                 |                  |                  |
|------------------|-----------------|-----------------|------------------|------------------|
| ACCLLT= ***** GX | CP.2FR= 000030  | C.RSV 000002    | FC.MLD= 000026   | IS.DAO= ***** GX |
| ADDMAL= ***** GX | CS.ABO= 000100  | C.STA 000007    | FC.PCT= 000030   | IS.SUC= ***** GX |
| AS\$CHK= 000000  | CS.BRO= 000002  | C.STS 000012    | FC.PWR= 000022   | IS\$RAR= 000000  |
| AS\$CPS= 000000  | CS.BUF= 000200  | C.URM 177776    | FC.RCE= 000002   | IS\$RDN= 000000  |
| AS\$PRI= 000000  | CS.CES= 000002  | C.XACP 000004   | FC.RCP= 000014   | I.PRM = ***** GX |
| AS\$TRP= 000000  | CS.CHN= 000010  | C.XID 000035    | FC.TJM= 000010   | KISAR6= ***** GX |
| CB.CCB= 000002   | CS.CMP= 000200  | C.XLEN 000044   | FC.XCP= 000012   | K\$CNT= 177546   |
| CB.DDM= 000040   | CS.DCR= 000400  | C.XPLI 000040   | FC.XME= 000000   | K\$CLSR= 177546  |
| CB.DLC= 000020   | CS.DEF= 000004  | C.XPT 000034    | FLSHIO= ***** GX | K\$SLDC= 000000  |
| CB.RDB= 000004   | CS.DEV= 000002  | C.XSVC 000042   | FS.AST= 000000   | K\$SLPS= 000074  |
| CB.SDB= 000010   | CS.DIS= 000040  | C.XTC 000037    | FS.CIB= 002000   | LA.ACK= 100000   |
| CB.SLI= 000100   | CS.ENA= 000001  | C.X25 000036    | FS.CRA= 001000   | LA.CRS= 020000   |
| CB.XLB= 000001   | CS.ENB= 000020  | DISCMP 000520RG | FS.DIS= 013000   | LA.MSK= 170000   |
| CCBGT= ***** GX  | CS.ERR= 100000  | D\$BUG= 177514  | FS.DVC= 001000   | LA.NAK= 110000   |
| CCBRT= ***** GX  | CS.FTL= 001000  | D\$ISK= 000000  | FS.ENB= 012000   | LA.NMS= 010000   |
| CC.LLC= 000200   | CS.HCR= 000001  | D\$LL1= 000001  | FS.EXI= 001000   | LA.RES= 040000   |
| CE.ABO= 100362   | CS.HFE= 002000  | D\$SYNC= 000000 | FS.GET= 006000   | LA.WND= 004000   |
| CE.DAO= 100346   | CS.LST= 040000  | D\$SYNM= 000000 | FS.HLT= 000000   | LD\$LP = 000000  |
| CE.DIS= 100366   | CS.MTL= 004000  | ER\$ABM= 000010 | FS.INI= 000000   | LF.DRD= 000004   |
| CE.ERR= 100370   | CS.RNG= 000010  | ER\$ABO= 000046 | FS.KIL= 000000   | LF.FRC= 000001   |
| CE.ILN= 100350   | CS.ROV= 000004  | ER\$ABT= 000011 | FS.LCL= 100000   | LF.HFO= 000010   |
| CE.LTD= 100356   | CS.RSN= 010000  | ER\$ACC= 000042 | FS.LTM= 001000   | LF.HMF= 000040   |
| CE.MDP= 100372   | CS.SHU= 000001  | ER\$CDI= 000052 | FS.MNT= 004000   | LF.RSC= 000020   |
| CE.NTE= 100361   | CS.SID= 000002  | ER\$COM= 000047 | FS.MSN= 014000   | LF.IRD= 000002   |
| CE.RTE= 100376   | CS.STR= 000004  | ER\$FMT= 000005 | FS.REA= 001000   | LF.MMF= 000200   |
| CE.SRC= 100364   | CS.SUC= 000001  | ER\$MLB= 000006 | FS.RET= 000000   | LF.MSF= 000100   |
| CE.STP= 100352   | CS.TMO= 020000  | ER\$NNF= 000012 | FS.REZ= 003000   | LS.DLS= 100000   |
| CE.TMF= 100354   | CS.XUR= 000004  | ER\$NOD= 000002 | FS.RLB= 002000   | LS.FCC= 000004   |
| CE.TMO= 100374   | CV\$MSK= 000003 | ER\$NSL= 000013 | FS.RNG= 011000   | LS.FCO= 000001   |
| CE.UNS= 100344   | CV\$31 = 000001 | ER\$NSR= 000003 | FS.RST= 000000   | LS.FCI= 000002   |
| CF.CHN= 000001   | CV\$32 = 000000 | ER\$RES= 000001 | FS.RTN= 001000   | LS.ILS= 100000   |
| CF.EOM= 000004   | CV\$40 = 000002 | ER\$STA= 000051 | FS.SET= 005000   | LS.MAK= 000020   |
| CF.HDR= 000020   | CX.GDQ= 000001  | ER\$UOB= 000004 | FS.SFC= 005000   | LS.MNK= 000040   |
| CF.LB = 100000   | CX.REM= 000020  | ESNBR 000014    | FS.SFR= 006000   | LS.RES= 000360   |
| CF.LIN= 000002   | CX.REQ= 000002  | ESNBS 000020    | FS.SFS= 004000   | LS.RSV= 000300   |
| CF.SOM= 000010   | CX.RUI= 000040  | ESNCR 000034    | FS.SPW= 040000   | LT.CCA= 000020   |
| CF.SYN= 000040   | CX.SMC= 000010  | ESNCS 000036    | FS.STM= 000000   | LT.DIR= 000010   |
| CF.TRN= 000100   | CX.UNL= 000004  | ESNIC 000044    | FS.STP= 002000   | LT.LCL= 000001   |
| CL\$MFL= 000010  | C\$QRE= 000400  | ESNLEN 000050   | FS.STR= 001000   | LT.LPL= 000002   |
| CL\$SFL= 000004  | C\$RSH= 177564  | ESNLLA 000012   | FS.TRM= 003000   | LT.NOT= 000040   |
| CL\$TYP= 000001  | C.ADD 000034    | ESNLNK 000000   | FS.WLB= 001000   | LT.RSU= 000200   |
| CL.MU1= 000001   | C.BID 000003    | ESNML 000040    | FS.XKL= 002000   | LT.SLI= 000004   |
| CL.MU2= 000002   | C.BUF 000014    | ESNMR 000024    | FS.XOF= 010000   | LT.TDA= 000100   |
| CL.RES= 177774   | C.BUF1 000014   | ESNMS 000030    | FS.XON= 007000   | L\$ASG= 000000   |
| CM.CIR= 000002   | C.BUF2 000024   | ESNNOD 000002   | FS.ZER= 002000   | L\$DRV= 000000   |
| CM.CDN= 000200   | C.CNT 000020    | ESNRT 000042    | FS\$LVL= 000001  | L\$PT1= 000001   |
| CM.FMT= 100000   | C.CNT1 000020   | ESNRT 000005    | G\$TTP= 000000   | L\$T1R= 000000   |
| CM.HRD= 000002   | C.CNT2 000030   | ESNSEG 000010   | G\$TSS= 000000   | L.CSTA 000037    |
| CM.LIN= 000000   | C.FLG 000022    | ESNTJM 000046   | G\$TTK= 000000   | L.CTR 000074     |
| CM.LOD= 000001   | C.FLG1 000022   | ESNUSE 000004   | G\$WRD= 000000   | L.DCR 000100     |
| CM.XLO= 000004   | C.FLG2 000032   | ESSTR 000006    | IE.DAO= ***** GX | L.FLAG 000014    |
| CPYCNC 000000R   | C.FNC 000010    | ES\$XPR= 000000 | IE.NRJ= ***** GX | L.ILSQ 000052    |
| CP.DCF= 000040   | C.LIN 000006    | FC.CCP= 000020  | IE.URJ= ***** GX | L.ILTT 000066    |
| CP.HDL= 000007   | C.LNK 000000    | FC.CTL= 000006  | IN.DAT= 000400   | L.LDA 000032     |
| CP.PS = 177400   | C.MOD 000011    | FC.KCP= 000016  | IN.ILS= 000001   | L.LIA 000034     |
| CP.PSI= 000200   | C.NSP 000004    | FC.KIL= 000004  | IODUN = ***** GX | L.LLA 000002     |
| CP.XCF= 000100   | C.PRO 000042    | FC.MAN= 000024  | IOSUC = ***** GX | L.LNG 000124     |

```

113 .SBTTL Connect received acknowledgement
114
115 ;+
116 ***-SLICRA-Connect received acknowledgement
117
118 This routine is called to acknowlege the receipt of an incoming
119 connect.
120
121 Inputs:
122 R4 = Address of CCB
123 C.NSP - Physical address of LLT (LLA)
124 C.FLG2 - ULA
125 C.STS - Acknowledgement status
126 C.STA - Source PDV index
127 R5 = Address of database descriptor
128
129 Registers modified:
130 R0, R1, R2, R3
131
132 .PSECT
133
134 SLICRA: CALLE CHKLLA ; Check for a valid LLA
135 BCS 30$; If CS, invalid LLA supplied
136
137 MOVWB C.FLG2(R4),L,ULA(R3)
138 MOVWB C.STA(R4),L,PDVC(R3)
139 MOVWB C.STA(R4),L,PDVD(R3)
140 MOV C.STS(R4),-(SP) ; Save acknowledgement status
141 CALL RLSCI ; Release the CI resources
142 MOVWB (SP)+,R1 ; Recover acknowledgement status
143 BMI 10$; If MI, reject the connection now
144
145 CALLE ACKCI ; Acknowledge the CI
146 RETURN
147
148 10$: CALL @CSBG7 ; Allocate a combined CCB and small data buffer
149 BCS 20$; If CS, none available
150
151 MAPLLT ; Recover mapping to the LLT
152 CLR C.NSP(R4) ; No logical link for this message
153 MOVWB #NT$DC,C.RSV(R4); This will be a disconnect confirm
154 MOVWB L.CHN(R3),C.LIN(R4)
155
156 NEG R1 ; Form NSP reason code
157 MOV R1,NSERRC(R5) ; Set up reason code for DC response
158 CLR NSDLA(R5) ; Clear out source link address
159 CALLE SENDDC ; Send the DC to break the link
160
161 20$: CALLR REMLNK ; Remove the logical link data structures
162
163 30$: CALLR RLSCI ; Release CI resources

```



```

610 .SBTIL Find node mapping by address
611 :+
612 **--FNDADD-Find node mapping by node address
613 :
614 Scan the remote name blocks to find a match on the specified node
615 address.
616 :-
617 Inputs:
618 R5 = Address of database descriptor
619 :
620 Outputs:
621 'C' Clear - Valid address mapping found
622 'C' Set - No valid address mapping found
623 :
624 Registers modified:
625 R1, R3
626 FNDADD: MOV @DECPT,R3 ; Point to the DEC home block
627 MOV D$RNN(R3),R3 ; Point to remote node listhead
628
629 10$: MOV (R3),-(SP) ; Get address of next block
630 CALL @CEACC ; Gain access to the block
631 MOV (SP)+,R3 ; Retrieve mapped address
632 BEQ 20$; If EQ, no more
633
634 CMP R,ADD(R3),$WORK ; Match on node address?
635 BNE 10$; If NE, no ... keep looking
636
637 ADD #R.NAM,R3 ; Point to remote node name
638 MOV #$WORK+2,R0 ; Point to output workspace
639 CALL NODIMG ; Convert to image field
640
641 20$: TST (PC)+ ; Indicate success
642 SEC ; Indicate failure
643 RETURN
644

```

SESSL111S CREATED BY MACRO ON 28-JUN-85 AT 19:59

PAGE 2 H 10

MACRO CROSS REFERENCE

CREF 04.00

MACRO NAME REFERENCES

|         |       |      |
|---------|-------|------|
| CALLE   | #6-44 |      |
| CCBDF\$ | #6-45 | 6-49 |
| CNBDF\$ | #6-45 | 6-53 |
| DHBD\$  | #6-46 | 6-55 |
| ECDD\$  | #6-45 | 6-50 |
| MAP     | #6-44 |      |
| MAPLL T | #6-44 |      |
| MSGDF\$ | #6-45 | 6-52 |
| PDVDF\$ | #6-45 | 6-51 |
| RESRG   | #6-44 |      |
| RNBDF\$ | #6-45 | 6-54 |
| SAVRG   | #6-44 |      |
| SLIDF\$ | #6-45 | 6-48 |

```

381 .SBTTL Convert RAD50 to ASCII
382 ;+
383 ;**--C5TA--Convert RAD50 to ASCII
384 ;
385 ; Convert 3 RAD50 character to ASCII.
386 ;
387 ;Inputs:
388 ; R1 = RAD50 word to be converted
389 ; R2 = Address of next byte in output buffer
390 ;
391 ;Outputs:
392 ; R2 = Address of next byte in output buffer
393 ;
394 ;Registers modified:
395 ; R0, R1
396 ;
397 .PSECT $HIGH
398
399 C5TA:: CALL CVTC ; Convert first character
400 MOV R0,-(SP) ; and save it
401 CALL CVTC ; Convert second character
402 MOV R0,-(SP) ; and save it
403 CALL CVTC ; Convert third character
404
405 MOVB R0,(R2)+ ; Store it
406 MOVB (SP)+,(R2)+ ; and second character
407 MOVB (SP)+,(R2)+ ; and first character
408 RETURN

```

```

878 .SBTTL Flush queue of I/O requests
879
880 ;+
881 ***-FLSHRQ-Flush queue of I/O requests
882
883 Complete a queue of I/O requests in error without removing the LUN
884 interlock.
885
886 Inputs:
887 R3 = Pointer to I/O packet queue
888 R5 = Address of database descriptor
889
890 Registers modified:
891 R0, R1, R3, R4
892
893 000306 011346
894 000310 012700 000000C
895 000314 005001
896
897 .IF DF N$$BUF
898
899 TST I.FCN(R3) ; Is this a buffered I/O request?
900 BPL 10$; If PL, no
901 CLR I.PRM+12(R3) ; Indicate no data to be moved
902 CALL @QUEBF ; Complete the request
903 BR 20$; Continue in common code
904
905 10$:
906 .ENDC
907
908 000316 005063 000016G
909 000322
910 000326 012603
911 000330 001366
912 000332
913
914 CLR I.PRM+16(R3) ; Clear RMS record locking flag
915 CALL @IOFIN ; Complete the request
916 MOV (SP)+,R3 ; Get address of next I/O packet
917 BNE FLSHRQ ; Loop till all done
918 RETURN

```

1347  
 1348  
 1349  
 1350  
 1351  
 1352  
 1353  
 1354  
 1355  
 1356  
 1357  
 1358  
 1359  
 1360  
 1361  
 1362 002670  
 1363 002676  
 1364 002702 016403 000006  
 1365 002706  
 1366 002716  
 1367 002722  
 1368 002726  
 1369 002732  
 1370 002740

```

.SBTTL Reject a logical link internally
+
**--REJECT--Reject a logical link internally
 Reject a logical link because the network software is unable
 to handle the request.
-
Inputs:
 R1 = Reason for the reject
 R4 = Address of pending connect CCB
 R5 = Address of database descriptor
Registers modified:
 R1, R4
REJECT::SAVRG <R0,R2,R3> ; Save some registers
 SAVMAP ; Save current mapping
 MOV C.LIN(R4),R3 ; Get physical address of LLT
 CALLE ACCLLT ; Gain access to LLT
 CALL RLSCI ; Release the connect resources
 CALL BRKLNK ; Break the logical link
 RESMAP ; Restore mapping
 RESRG <R3,R2,R0> ; Restore registers
 RETURN

```

|                  |                 |                  |                  |                  |
|------------------|-----------------|------------------|------------------|------------------|
| MF.DAT= 000000   | NF\$BLK= 000100 | N\$FNC 000006    | OF.PRO= 000040   | RT\$OFF= 000001  |
| MO\$SAC= 000016  | NF\$DMO= 000010 | N\$GENQ 000052   | OF.RLU= 000100   | RT\$ON = 000000  |
| MO\$SPR= 000012  | NF\$MOU= 000040 | N\$GTM 000015    | OF.SMC= 000200   | R\$DER= 000000   |
| MO\$SSV= 000014  | NF\$RST= 000002 | N\$HIGH 000033   | OP\$INI= 000000  | R\$K11= 000001   |
| MO\$25A= 000006  | NF\$SCN= 000020 | N\$LLT 000026    | OP\$TER= 000001  | R\$SND= 000000   |
| MO\$25P= 000002  | NF\$SHU= 000004 | N\$LLTM 000024   | O.FLG 000003     | R\$11M= 000000   |
| MO\$25S= 000004  | NF\$TIM= 000200 | N\$LVC 000036    | O.LEN 000012     | R\$11S= 000000   |
| MO\$29S= 000010  | NM\$ARA= 176000 | N\$MBXQ 000050   | O.LNK 000000     | R.ADD 000010     |
| MVFBF = ***** GX | NM\$NOD= 001777 | N\$PLLT 000030   | O.MXC 000004     | R.FLAG 000012    |
| M\$HIGH= 000003  | NOOP 003506R    | N\$SLA 000016    | O.NAM 000006     | R.LEN 000014     |
| M\$CRB= 000124   | NO.DTR= 000077  | N\$SNOD 000012   | O.TYP 000002     | R.LNK 000000     |
| M\$CRX= 000000   | NO.FAL= 000021  | N\$TIM 000004    | PH\$HDE= 000004  | R.NAM 000002     |
| M\$FCS= 000000   | NO.FAI= 000001  | N\$VBC 000010    | PH\$LOC= 000002  | SAVOPT 003276RG  |
| M\$MGE= 000000   | NO.NCU= 000023  | N\$SACC= 000001  | PH\$MTS= 000003  | SC\$OFF= 000001  |
| M\$MUP= 000000   | NO.RTL= 000022  | N\$SACK= 000011  | PH\$UMP= 000000  | SC\$ON = 000000  |
| M\$NET= 000000   | NO.TAS= 000000  | N\$SEVL= 000001  | PH\$WCS= 000001  | SC\$RST= 000003  |
| M\$QVR= 000000   | NO.TCL= 000017  | N\$SHDR= 000007  | PNTCCB 002516RG  | SC\$SHU= 000002  |
| M\$3100= 000000  | NO.TCI= 000005  | N\$SLDV= 000001  | PROIMG 002554R   | SENDI= ***** GX  |
| M\$3101= 000001  | NO.TLK= 000020  | N\$SMLL= 000001  | PRONAM 002576R   | SPRST= ***** GX  |
| M\$3102= 000002  | NSTATE 003510RG | N\$SMOV= 000010  | P\$P45= 000000   | ST\$CC = 000004  |
| M\$3103= 000003  | N\$SDON= 000000 | N\$SNCT= 000001  | P\$SWRD= 000000  | ST\$CIR= 000006  |
| M.MAIL 000014    | N\$SSDI= 000002 | N\$SOVR= 000022  | P.HDR = ***** GX | ST\$CIS= 000002  |
| M.MAX 000011     | N\$SWDC= 000004 | N\$SPEM= 000001  | Q\$SOPT= 000010  | ST\$DAT= 000010  |
| M.MBL 000020     | NT\$AKD= 000020 | N\$SSES= 000001  | REJECT 002670RG  | ST\$DIP= 000012  |
| M.NAST 000007    | NT\$AKI= 000022 | N.CAC 000120     | REMLNK 002742RG  | ST\$PND= 000014  |
| M.NEXT 000002    | NT\$CC = 000016 | N.CACC 000116    | RETRES 003074RG  | SV\$DUM= 000001  |
| M.RESP 000016    | NT\$CON= 000000 | N.CBL = 000142   | RETTB 003134R    | SV\$LOA= 000000  |
| M.SPA 000012     | NT\$CTL= 000000 | N.CDA 000142     | RE\$ADC= 000004  | \$SABO= 000022   |
| M.TASK 000004    | NT\$DAT= 000002 | N.CDAC 000140    | RE\$ADF= 000017  | \$SACC = 000002  |
| M.USE 000010     | NT\$DC = 000012 | N.CDDS 000070    | RE\$ADR= 000007  | \$SBUF = 000026  |
| NC.FM0= 000000   | NT\$DIS= 000014 | N.CDEV 000062    | RE\$BLK= 000010  | \$SCNR = 000000  |
| NC.FM1= 000001   | NT\$DLS= 000006 | N.CID 000064     | RE\$CAF= 000014  | \$SCON = 000000  |
| NC.FM2= 000002   | NT\$ILS= 000010 | N.CIDC 000062    | RE\$DAT= 000001  | \$SDAT = 000002  |
| NE\$ABM= 000010  | NT\$IMS= 000002 | N.CPS 000106     | RE\$DRP= 000016  | \$SDIS = 000020  |
| NE\$ABO= 000046  | NT\$INT= 000004 | N.CPSC 000104    | RE\$LDI= 000013  | \$SDRQ = 000012  |
| NE\$ACC= 000042  | NT\$RET= 000032 | N.CTL 000000     | RE\$LSN= 000012  | \$SDSR = 000006  |
| NE\$ACT= 000042  | NT\$ROU= 000024 | N.CUIC 000066    | RE\$NML= 000001  | \$SGLN = 000024  |
| NE\$COM= 000047  | NT\$RTR= 000030 | N.CUNI 000064    | RE\$OPE= 000004  | \$SINT = 000004  |
| NE\$FCF= 000050  | NT\$TSP= 000026 | N.DDE 000010     | RE\$OPR= 000000  | \$SIRQ = 000016  |
| NE\$FMT= 000005  | NT.ABO= 000005  | N.DDEC 000006    | RE\$RCV= 000001  | \$SMRQ = 000010  |
| NE\$GEN= 000007  | NT.ABT= 000004  | N.DFM 000004     | RE\$SED= 000011  | \$SNOT = 000010  |
| NE\$IFC= 000030  | NT.CON= 000001  | N.DGP 000006     | RE\$SKW= 000006  | \$SNTIF= 000001  |
| NE\$ILS= 000043  | NT.DSC= 000003  | N.DNM 000012     | RE\$STA= 000002  | \$SPBOM= 000040  |
| NE\$IMG= 000042  | NT.EVT= 000006  | N.DNMC 000012    | RE\$SUM= 000003  | \$SPEOM= 000100  |
| NE\$MLB= 000006  | NT.INT= 000002  | N.DOT 000005     | RE\$SYN= 000000  | \$SPMOM= 000000  |
| NE\$NNF= 000012  | NT.MOP= 000010  | N.DUS 000010     | RE\$TME= 000021  | \$SPMSG= 000200  |
| NE\$NOD= 000002  | NT.NSP= 000010  | N.SDE 000042     | RE\$TMD= 000000  | \$SPSEG= 000100  |
| NE\$NSD= 000003  | NT.VFY= 000007  | N.SDEC 000040    | RE\$TMR= 000020  | \$SREJ = 000004  |
| NE\$NSL= 000013  | N\$ACQ 000000   | N.SEGZ 000002    | RE\$UPT= 000002  | \$SSND = 000006  |
| NE\$NSR= 000003  | N\$ACTL 000032  | N.SFM 000036     | RE\$URE= 000003  | \$SSNI = 000014  |
| NE\$RES= 000001  | N\$CIR 000034   | N.SGP 000040     | RE\$VER= 000005  | \$STMDA= 000002  |
| NE\$SSR= 000000  | N\$DLA 000020   | N.SND 000030     | RE\$VRQ= 000015  | \$S\$WRG= 000000 |
| NE\$SSS= 000045  | N\$DLY 000014   | N.SNM 000046     | RF.LOO= 100000   | \$S\$YSZ= 007600 |
| NE\$TCN= 000040  | N\$ELEN 000054  | N.SNMC 000044    | RLSCI 003144RG   | S.EABL= 100200   |
| NE\$TCO= 000006  | N\$ENC 000042   | N.SOT 000037     | RMVWND 003176RG  | S.EABM= 100370   |
| NE\$TPA= 000010  | N\$ERRC 000022  | N.SUS 000042     | RSNDDI= ***** GX | S.EABO= 100367   |
| NE\$UOB= 000004  | N\$FLG 000005   | OBJHD = ***** GX | RT\$INI= 000002  | S.EABS= 100202   |

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44

```
.TITLE SESTCB - TCB copy routine for RSX-11M
.IDENT /V05.00/
.ENABL LC
```

```
: Copyright (C) 1982, 1983, 1985 by
: Digital Equipment Corporation, Maynard, MASS.
```

```
: This software is furnished under a license for use only on a
: single computer system and may be copied only with the
: inclusion of the above copyright notice. This software, or
: any other copies thereof, may not be provided or otherwise
: made available to any other person except for use on such
: system and to one who agrees to these license terms. Title
: to and ownership of the software shall at all times remain
: in DEC.
```

```
: The information in this document is subject to change without
: notice and should not be construed as a commitment by Digital
: Equipment Corporation.
```

```
: DEC assumes no responsibility for the use or reliability of
: its software on equipment which is not supplied by DEC.
```

```
: Module description:
```

```
: ACP user TCB copying routine
```

```
: Distributed systems software engineering
```

```
: Ident history:
```

```
: 4.00 07-NOV-83
: DECNET-11M V4.0
: DECNET-11M-PLUS V2.0
:
: 5.00 22-JUL-85
: DECnet-11M/S V4.2
: DECnet-11M-Plus V3.0
: DECnet-Micro/RSX V1.0
```

```
.MCALL SAVRG,RESRG
```

SESTIM - Session control timer MACRO V05.03b Friday 28-Jun-85 20:01 <sup>H 16</sup> Page 8-1  
Session control timer service action routines

```
166 000220 30$: CALLR TRMNET ; Process network disconnect state
167
168 .DSABL LSB
169
170 000001 .END
```

SESTIM - Session control timer MACRO V05.03b Friday 28-Jun-85 20:01 <sup>I 16</sup> Page 8-2  
Symbol table



SESDSP - Session control dispat MACRO V05.03b Friday 28-Jun-85 19:56<sup>I 1</sup> Page 15-3  
Symbol table

Size of work file: 22605 Words ( 89 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:27.32  
SY:SESDSP11S.V2,[131,134]SESDSP11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCS/PA:1,[131,10]V2,SESDSP

|                  |                 |                     |                  |                    |
|------------------|-----------------|---------------------|------------------|--------------------|
| N\$SNOD 000012   | RE\$CAF= 000014 | SC\$RST= 000003     | T\$NLEN 000010   | ZF.KMX= 000020     |
| N\$TIM 000004    | RE\$DAT= 000001 | SC\$SHU= 000002     | T\$NNUL 000002   | ZF.LLC= 000004     |
| N\$VCB 000010    | RE\$DRP= 000016 | SESINI 000000RG 002 | T\$NOPL 000006   | ZF.LMC= 000100     |
| N\$ACC= 000001   | RE\$LDT= 000013 | SE\$SDV= 051516     | T\$NRNI 000042   | ZF.MAN= 020000     |
| N\$ACK= 000011   | RE\$LSN= 000012 | SV\$DUM= 000001     | T\$NRPL 000005   | ZF.MFL= 000010     |
| N\$EVL= 000001   | RE\$NML= 000001 | SV\$LOA= 000000     | T\$NRUL 000007   | ZF.MTM= 000400     |
| N\$HDR= 000007   | RE\$OPE= 000004 | S\$WRG= 000000      | T\$NVR 000001    | ZF.MUX= 000040     |
| N\$LDV= 000001   | RE\$OPR= 000000 | S\$YSZ= 007600      | T\$RPR 000040    | ZF.PSE= 002000     |
| N\$MLL= 000001   | RE\$RCV= 000001 | TKTCB = ***** GX    | T\$SVC 000034    | ZF.SLI= 010000     |
| N\$MOV= 000010   | RE\$SED= 000011 | TTNS = ***** GX     | T\$T5 000030     | ZF.TIM= 000200     |
| N\$NCT= 000001   | RE\$SKW= 000006 | T\$FLAG 000044      | T\$T6 000032     | ZF.X3P= 000000     |
| N\$OVR= 000022   | RE\$STA= 000002 | T\$LIF 000013       | T\$KMG= 000000   | ZS.ASN= 100000     |
| N\$PEM= 000001   | RE\$SUM= 000003 | T\$LIFL 000013      | T\$MIN= 000000   | ZS.BSY= 140000     |
| N\$SES= 000001   | RE\$SYN= 000000 | T\$LILO 000013      | US.MDM= ***** GX | Z.AVL 000014       |
| OP\$INI= 000000  | RE\$TME= 000021 | T\$LILO 000013      | US.MNT= ***** GX | Z.DAT 000016       |
| OP\$TER= 000001  | RE\$TMO= 000000 | T\$LIN 000000       | US.OFL= ***** GX | Z.DSP 000000       |
| PDVID = ***** GX | RE\$TMR= 000020 | T\$LIPL 000006      | U.ACP = ***** GX | Z.FLG 000010       |
| PDVTA = ***** GX | RE\$UPT= 000002 | T\$LLD 000012       | U.STS = ***** GX | Z.LEN = 000016     |
| PH\$HDE= 000004  | RE\$URE= 000003 | T\$LLDC 000045      | U.ST2 = ***** GX | Z.LLN 000006       |
| PH\$LOC= 000002  | RE\$VER= 000005 | T\$LLDL 000012      | U.VCB = ***** GX | Z.MAP 000020       |
| PH\$MTS= 000003  | RE\$VRQ= 000015 | T\$LLDD 000012      | V\$CTR= 001000   | Z.NAM 000004       |
| PH\$UMP= 000000  | RT\$INI= 000002 | T\$LLDS 000012      | WORD1 = 000200   | Z.PCB 000012       |
| PH\$WCS= 000001  | RT\$OFF= 000001 | T\$LLDN 000046      | WORD2 = 000000   | Z.SCH 000007       |
| P\$P45= 000000   | RT\$ON = 000000 | T\$LOPR 000002      | X\$DBT= 000000   | \$CALLX= ***** GX  |
| P\$WRD= 000000   | R\$DER= 000000  | T\$LTCL 000024      | ZF.COU= 001000   | \$CLIN= ***** GX   |
| Q\$OPT= 000010   | R\$K11= 000001  | T\$LTIM 000026      | ZF.DDM= 000001   | \$ENCODE= ***** GX |
| RDBSZ = ***** GX | R\$SND= 000000  | T\$LTNR 000014      | ZF.DIA= 004000   | \$ESDB= ***** GX   |
| RE\$ADC= 000004  | R\$11M= 000000  | T\$LTNS 000020      | ZF.DLC= 000002   | \$SESPD= ***** GX  |
| RE\$ADR= 000017  | R\$11S= 000000  | T\$NAPL 000004      | ZF.DVP= 100000   | \$UCB = ***** GX   |
| RE\$ADR= 000007  | SC\$OFF= 000001 | T\$NFE 000000       | ZF.INI= 040000   | \$XPTIN= ***** GX  |
| RE\$BLK= 000010  | SC\$ON = 000000 |                     |                  |                    |

. ABS. 000216 000 (RW,I,GBL,ABS,OVR)  
000000 001 (RW,I,LCL,REL,CON)  
\$HIGH 000412 002 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 94  
Work file writes: 73  
Size of work file: 20412 Words ( 80 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLJS

Elapsed time: 00:00:19.52  
SY:SESINI11S.V2,[131,134]SESINI11S/CR/-SP=SY:[1,1]RSXCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMC/PA:1,[131,10]V2,SESINI

SESMN - Session control main to MACRO V05.03b Friday 28-Jun-85 19:57<sup>13</sup> Page 7  
Define local macros

59  
60  
61  
62  
63  
64

.SBTTL Define local macros

.MACRO XQT,DPB  
JSR R5,\$XQTDR  
.WORD DPB  
.ENDM XQT

SESMN - Session control main to MACRO V05.03b Friday 28-Jun-85 19:57<sup>13</sup> Page 8  
Session control main logic

\*\*FILE\*\*ID\*\*SESPRO

! 4

|          |            |          |          |          |        |      |
|----------|------------|----------|----------|----------|--------|------|
| SSSSSSSS | EEEEEEEEEE | SSSSSSSS | PPPPPPPP | RRRRRRRR | 000000 |      |
| SSSSSSSS | EEEEEEEEEE | SSSSSSSS | PPPPPPPP | RRRRRRRR | 000000 |      |
| SS       | EE         | SS       | PP PP    | RR RR    | 00 00  |      |
| SS       | EE         | SS       | PP PP    | RR RR    | 00 00  |      |
| SS       | EE         | SS       | PP PP    | RR RR    | 00 00  |      |
| SS       | EE         | SS       | PP PP    | RR RR    | 00 00  |      |
| SSSSSS   | EEEEEEEE   | SSSSSS   | PPPPPPPP | RRRRRRRR | 00 00  |      |
| SSSSSS   | EEEEEEEE   | SSSSSS   | PPPPPPPP | RRRRRRRR | 00 00  |      |
| SS       | EE         | SS       | PP       | RR RR    | 00 00  |      |
| SS       | EE         | SS       | PP       | RR RR    | 00 00  |      |
| SS       | EE         | SS       | PP       | RR RR    | 00 00  |      |
| SS       | EE         | SS       | PP       | RR RR    | 00 00  |      |
| SSSSSSSS | EEEEEEEEEE | SSSSSSSS | PP       | RR RR    | 000000 | .... |
| SSSSSSSS | EEEEEEEEEE | SSSSSSSS | PP       | RR RR    | 000000 | .... |

|        |        |          |
|--------|--------|----------|
| 11     | 11     | SSSSSSSS |
| 11     | 11     | SSSSSSSS |
| 1111   | 1111   | SS       |
| 1111   | 1111   | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SSSSSS   |
| 11     | 11     | SSSSSS   |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 11     | 11     | SS       |
| 111111 | 111111 | SSSSSSSS |
| 111111 | 111111 | SSSSSSSS |

```

450 .SBTTL Get optional data from message
451 ;+
452 **-GETOPT-Get optional data from message
453 ;
454 Copy the optional data from the message to the optional data buffer
455 within the task.
456 -
457 Inputs:
458 R2 = Pointer to optional data field count
459 R3 = Virtual address of LLT
460 R4 = Address of CCB
461 R5 = Address of database descriptor
462 ;
463 Outputs:
464 'C' Clear - Optional data copied successfully
465 'C' Set - Message format error - optional data field too long
466 ;
467 Registers modified:
468 R0, R1, R2
469 ;
470 001320 112201 GETOPT: MOVB (R2)+,R1 ; Get byte count from message
471 001322 022701 CMP #16.,R1 ; Is it too long?
472 001326 103410 BLO 20$; If L0, yes (C-bit set)
473 ;
474 001330 010167 MOV R1,$OPLNG ; Save length of optional data
475 001334 001405 BEQ 20$; If EQ, none
476 001336 012700 MOV #$OPDAT,R0 ; Point to optional data buffer
477 001342 112220 10$: MOVB (R2)+,(R0)+ ; Copy the data
478 001344 SOB R1,10$; ...
479 ;
480 001350 20$: RETURN

```

MACRO CROSS REFERENCE

CREF 04.00

MACRO NAME REFERENCES

|         |         |        |         |        |         |        |         |        |         |        |
|---------|---------|--------|---------|--------|---------|--------|---------|--------|---------|--------|
| CALL    | 8-114   | 9-177  | 9-188   | 9-200  | 9-201   | 9-206  | 9-222   | 9-229  | 9-233   | 10-271 |
|         | 11-299  | 11-318 | 11-331  | 12-365 | 12-371  | 12-386 | 13-435  | 16-367 |         |        |
| CALLE   | #6-44   | 8-118  | 9-216   | 9-237  | 10-264  | 11-302 | 11-316  | 11-320 | 11-343  | 11-347 |
|         | 12-368  | 12-389 | 13-419  | 16-557 | 16-569  | 18-624 |         |        |         |        |
| CALLR   | 7-70    |        |         |        |         |        |         |        |         |        |
| CALLX   | #8-118  | 8-118  | #9-216  | 9-216  | #9-237  | 9-237  | #10-264 | 10-264 | #11-302 | 11-302 |
|         | #11-316 | 11-316 | #11-320 | 11-320 | #11-343 | 11-343 | #11-347 | 11-347 | #12-368 | 12-368 |
|         | #12-389 | 12-389 | #13-419 | 13-419 | #16-557 | 16-557 | #16-569 | 16-569 | #18-624 | 18-624 |
| CCBDF\$ | #6-45   | 6-47   |         |        |         |        |         |        |         |        |
| COUNT\$ | #6-44   | 9-197  | 9-198   | 11-310 | 11-311  | 12-380 | 12-381  | 12-382 | 13-422  |        |
| CTRDF\$ | #6-45   | 6-51   |         |        |         |        |         |        |         |        |
| DHBD\$  | #6-46   | 6-53   |         |        |         |        |         |        |         |        |
| ECDD\$  | #6-45   | 6-48   |         |        |         |        |         |        |         |        |
| LLTDF\$ | #6-45   | 6-49   |         |        |         |        |         |        |         |        |
| MAP     | #6-44   | 8-101  | 9-186   | 18-621 |         |        |         |        |         |        |
| MAPLLT  | #6-44   |        |         |        |         |        |         |        |         |        |
| MSGDF\$ | #6-45   | 6-50   |         |        |         |        |         |        |         |        |
| RESRG   | #6-44   |        |         |        |         |        |         |        |         |        |
| RETURN  | 8-121   | 9-217  | 9-239   | 10-273 | 11-345  | 11-349 | 12-392  | 13-437 | 14-480  | 15-542 |
|         | 16-576  | 17-596 | 18-626  |        |         |        |         |        |         |        |
| SAVRG   | #6-44   |        |         |        |         |        |         |        |         |        |
| SOB     | 9-185   | 14-478 |         |        |         |        |         |        |         |        |
| XPTDF\$ | #6-45   | 6-52   |         |        |         |        |         |        |         |        |

|          |        |          |        |          |        |          |         |          |          |
|----------|--------|----------|--------|----------|--------|----------|---------|----------|----------|
| L.LNO    | 000026 | M\$SOVR= | 000000 | N\$SWDC= | 000004 | N\$PEM=  | 000001  | T\$NLEN  | 000010   |
| L.LPT    | 000065 | M.MAIL   | 000014 | NT\$AKD= | 000020 | N\$SESE= | 000001  | T\$NNUL  | 000002   |
| L.LSA    | 000030 | M.MAX    | 000011 | NT\$AKI= | 000022 | N.RAC    | 000070  | T\$NOPL  | 000006   |
| L.LSFC   | 000046 | M.MBL    | 000020 | NT\$CC=  | 000016 | N.RACC   | 000066  | T\$NNRI  | 000042   |
| L.LSFI   | 000044 | M.NAST   | 000007 | NT\$CON= | 000000 | N.RDE    | 000012  | T\$NNPL  | 000005   |
| L.LTT    | 000062 | M.NEXT   | 000002 | NT\$CTL= | 000000 | N.RDEC   | 000010  | T\$NNRUL | 000007   |
| L.MASQ   | 000070 | M.RESP   | 000016 | NT\$DAT= | 000002 | N.RFM    | 000006  | T\$NNVR  | 000001   |
| L.MAST   | 000073 | M.SPA    | 000012 | NT\$DC=  | 000012 | N.RGP    | 000010  | T\$RPRI  | 000040   |
| L.MASZ   | 000072 | M.TASK   | 000004 | NT\$DIS= | 000014 | N.RID    | 000034  | T\$SVC   | 000034   |
| L.NIN    | 000020 | M.USE    | 000010 | NT\$DLS= | 000006 | N.RIDC   | 000032  | T\$T5    | 000030   |
| L.NXN    | 000016 | NC.FM0=  | 000000 | NT\$ILS= | 000011 | N.RND    | 000000  | T\$T6    | 000032   |
| L.NXTH   | 000010 | NC.FM1=  | 000001 | NT\$IMS= | 000012 | N.RNM    | 000016  | T\$SKMG= | 000000   |
| L.OPD    | 000103 | NC.FM2=  | 000002 | NT\$INT= | 000004 | N.RNMC   | 000014  | T\$SMIN= | 000000   |
| L.OPDL   | 000102 | NE\$ABM= | 000010 | NT\$RET= | 000032 | N.ROT    | 000007  | USRCC    | 000136RG |
| L.REM    | 000006 | NE\$ABO= | 000046 | NT\$ROU= | 000024 | N.RPS    | 000056  | USRCNF   | 000220RG |
| L.RFC    | 000050 | NE\$ACC= | 000042 | NT\$RTR= | 000030 | N.RPSC   | 000054  | USRDIS   | 000364RG |
| L.RLA    | 000004 | NE\$ACT= | 000042 | NT\$TSP= | 000026 | N.RQL    | 000110  | USRDSF   | 000312RG |
| L.RNO    | 000022 | NE\$COM= | 000047 | NT.ABO=  | 000005 | N.RUS    | 000012  | US\$CNF= | 000002   |
| L.RTQ    | 000060 | NE\$FCF= | 000050 | NT.ABT=  | 000004 | PNTCCB=  | *****   | US\$DIS= | 000006   |
| L.RTYD   | 000055 | NE\$FMT= | 000005 | NT.CON=  | 000001 | P\$P45=  | 000000  | US\$DON= | 000000   |
| L.RTYI   | 000057 | NE\$GEN= | 000007 | NT.DSC=  | 000003 | P\$WRD=  | 000000  | US\$DSC= | 000004   |
| L.SEC    | 000064 | NE\$IFC= | 000030 | NT.EVT=  | 000006 | Q\$OPT=  | 000010  | US\$EAC= | 000012   |
| L.SEGZ   | 000076 | NE\$ILS= | 000043 | NT.INT=  | 000002 | RMVLNK   | 000102R | US\$WDS= | 000010   |
| L.STA    | 000000 | NE\$IMG= | 000042 | NT.MOP=  | 000010 | RMVWND=  | *****   | VE.FAI=  | 177777   |
| L.TC     | 000042 | NE\$MLB= | 000006 | NT.NSP=  | 000010 | R\$SDER= | 000000  | VS.NPV=  | 000001   |
| L.TIC    | 000043 | NE\$NMF= | 000012 | NT.VF=   | 000007 | R\$SK11= | 000001  | VS.PRV=  | 000002   |
| L.TIPD   | 000013 | NE\$NOD= | 000002 | N\$ACQ   | 000000 | R\$SND=  | 000000  | VZ.NVD=  | 000000   |
| L.TIPI   | 000012 | NE\$NSD= | 000003 | N\$ACTL  | 000032 | R\$S11M= | 000000  | V\$CTR=  | 001000   |
| L.TMRD   | 000054 | NE\$NSL= | 000013 | N\$CJR   | 000034 | R\$S11S= | 000000  | WK.ACK=  | 000001   |
| L.TMRI   | 000056 | NE\$NSR= | 000003 | N\$DLA   | 000020 | ST\$CC=  | 000004  | WK.AST=  | 000200   |
| L.TYP    | 000001 | NE\$RES= | 000001 | N\$DLY   | 000014 | ST\$CIR= | 000006  | WK.DIS=  | 000010   |
| L.USA    | 000024 | NE\$SSR= | 000000 | N\$ELEN  | 000054 | ST\$CIS= | 000002  | WK.INT=  | 000020   |
| L.USTA   | 000036 | NE\$SSS= | 000045 | N\$ENC   | 000042 | ST\$DAT= | 000010  | WK.RCV=  | 000004   |
| L.VER    | 000015 | NE\$TCN= | 000040 | N\$ERRC  | 000022 | ST\$DIP= | 000012  | WK.SND=  | 000002   |
| L.WIND   | 000040 | NE\$TCO= | 000006 | N\$FLG   | 000005 | ST\$PND= | 000014  | WS.DIP=  | 000010   |
| MA.CI=   | 000040 | NE\$TPA= | 000010 | N\$FNC   | 000006 | S\$SWRG= | 000000  | WS.INT=  | 000002   |
| MA.DA=   | 000000 | NE\$UOB= | 000004 | N\$GENQ  | 000052 | S\$YSZ=  | 007600  | WS.KAS=  | 000004   |
| MA.IL=   | 000020 | NF\$BLK= | 000100 | N\$GIM   | 000015 | T\$FLAG  | 000044  | WS.STA=  | 000001   |
| MC.CC=   | 000040 | NF\$DMO= | 000010 | N\$HIGH  | 000033 | T\$LIF   | 000013  | W.CINT   | 000022   |
| MC.CI=   | 000020 | NF\$MOU= | 000040 | N\$LLT   | 000026 | T\$LIFL  | 000013  | W.CSND   | 000020   |
| MC.DC=   | 000100 | NF\$RST= | 000002 | N\$LLTM  | 000024 | T\$LIFO  | 000013  | W.CTL    | 000000   |
| MC.DI=   | 000060 | NF\$SCN= | 000020 | N\$LVC   | 000036 | T\$LIFS  | 000013  | W.KAST   | 000014   |
| MC.NC=   | 000000 | NF\$SHU= | 000004 | N\$MBXQ  | 000050 | T\$LIN   | 000000  | W.LLT    | 000004   |
| MC.RC=   | 000140 | NF\$TIM= | 000200 | N\$PLLT  | 000030 | T\$LIPS  | 000006  | W.LUN    | 000003   |
| MD.BM=   | 000040 | NM\$ARA= | 176000 | N\$SLA   | 000016 | T\$LLD   | 000012  | W.MBOX   | 000012   |
| MD.EM=   | 000100 | NM\$NOD= | 001777 | N\$SNOD  | 000012 | T\$LLDC  | 000045  | W.RCVQ   | 000024   |
| MD.ILS=  | 000040 | NO.DTR=  | 000077 | N\$TIM   | 000004 | T\$LLDD  | 000012  | W.SEGZ   | 000006   |
| MD.LM=   | 000020 | NO.FAL=  | 000021 | N\$VCB   | 000010 | T\$LLDQ  | 000012  | W.SNDQ   | 000016   |
| MF.ACK=  | 000004 | NO.FAT=  | 000001 | N\$SACC= | 000001 | T\$LLDS  | 000012  | W.STAT   | 000002   |
| MF.CTL=  | 000010 | NO.NCU=  | 000023 | N\$SACK= | 000011 | T\$LLEN  | 000046  | W.TMP    | 000010   |
| MF.DAT=  | 000000 | NO.RTL=  | 000022 | N\$SEVL= | 000001 | T\$LOPR  | 000002  | W.WBL    | 000026   |
| M\$SCRB= | 000124 | NO.TAS=  | 000000 | N\$SHDR= | 000007 | T\$LTCL  | 000024  | X\$SDBT= | 000000   |
| M\$SCRX= | 000000 | NO.TCL=  | 000017 | N\$SLDV= | 000001 | T\$LIIM  | 000026  | \$CALLX= | ***** GX |
| M\$SFCS= | 000000 | NO.TCI=  | 000005 | N\$SMLL= | 000001 | T\$LTPR  | 000014  | \$IOPKT= | ***** GX |
| M\$SMGE= | 000000 | NO.TLK=  | 000020 | N\$SMOV= | 000010 | T\$LTPS  | 000020  | \$MAIBX= | ***** GX |
| M\$SMJP= | 000000 | N\$SDON= | 000000 | N\$SNCT= | 000001 | T\$NAPL  | 000004  | \$OPDAT= | ***** GX |
| M\$SNET= | 000000 | N\$SDSI= | 000002 | N\$SOVR= | 000022 | T\$NFE   | 000000  | \$OPLNG= | ***** GX |

```

164 .SBTTL SLI outgoing connect request
165
166 ***-SLICON-SLI outgoing connect request
167
168 This routine is called to perform an outgoing connect request for
169 the system level interface.
170
171 Inputs:
172 R4 = Address of CCB
173 C.BUF1 - Connect request block
174 C.BUF2 - Optional receive data buffer
175 C.FLG2 - ULA
176 C.LIN - Flow control options
177 C.STA - Source PDV index
178 R5 = Address of database descriptor
179
180 SLICON: MAP C.BUF(R4) ; Map to the connect request block
181 MOV C.BUF+2(R4),R2 ; Get virtual address of block
182 MOV # $CNBLK,R1 ; Point to internal connect block
183 MOV C.CNT(R4),R0 ; Get size of connect block
184 10$: MOVB (R2)+(R1)+ ; Copy connect block to local buffer
185 SOB R0,10$; ...
186
187 MOVB C.LIN(R4),$FLOW ; Set up requested flow control options
188 CLR $LTM ; No long term timer support
189 MOVB #LT.SLI,$LTYPE ; This is a system level interface link
190
191 CALL SNDCON ; Transmit the connect request
192 BCS 100$; If CS, error
193 MOV $RCCB,R4 ; Recover initial CCB
194 MOV R4,L.ACC(R3) ; Save control CCB
195 MOVB C.FLG2(R4),L.ULA(R3)
196 MOVB C.STA(R4),L.PDVC(R3)
197 MOVB C.STA(R4),L.PDVD(R3)
198 RETURN
199
200 100$: MOV N$ERRC(R5),R1 ; Get the error code
201 CALLR XMEERR ; Complete the request in error

```



```

646 .SBTTL User disconnect completion
647
648 ;+
649 ;**~SLIDSC-User disconnect completion
650 ;
651 ; Complete the user disconnect request when the network disconnect
652 ; processing has finished.
653 ;
654 ; Inputs:
655 ; R3 = Virtual address of LLT
656 ; R5 = Address of database descriptor
657 ;
658 ; Registers modified:
659 ; R0, R1, R2, R4
660 SLIDSC::TST L.TIPI(R3) ; Are there any transmits outstanding?
661 BNE 10$; If NE, yes ... delay completion
662 CALL FLSXMT ; Flush pending transmits
663
664 CMPB #NS$DON,L.CSTA(R3)
665 BNE 10$; Wait until network processing is complete
666
667 MOVB #US$DON,L.USTA(R3)
668 MOV L.PCTL(R3),R4 ; Get address of control CCB
669 MOV #S.SSUC,R1 ; Disconnect has completed with success
670 CALL XMECM2 ; Complete the request
671 10$: RETURN

```

┌

22

SESSUB - Session control subrou MACRO V05.03b Friday 28-Jun-85 19:59 J 10  
Table of contents

```

410 .SBTTL Convert one RAD50 character
411
412 ;+
413 ;**--CVTC--Convert one RAD50 character
414 ;
415 ; Convert the next RAD50 character to ASCII.
416 ;
417 ; Inputs:
418 ; R1 = Packed RAD50 word
419 ;
420 ; Outputs:
421 ; R0 = Next character from word
422 ; R1 = Remaining RAD50 characters
423
424 000030 CVTC: .IF DF R$$EIS
425
426 CLR R0 ; Clear upper part
427 DIV #50,R0 ; Divide them up
428
429 .IFF
430
431 000030 010100 MOV R1,R0 ; Copy dividend
432 000032 012701 000050 MOV #50,R1 ; Set up divisor
433 000036 CALL @CEDIV ; Perform the division
434
435 .ENDC
436
437 000042 010146 MOV R1,-(SP) ; Save remainder
438 000044 010001 MOV R0,R1 ; Quotient to remainder
439 000046 012600 MOV (SP)+,R0 ; Recover remainder
440 000050 001412 BEQ 30$; If ZERO, blank
441
442 000052 020027 000033 CMP R0,#33 ; Test middle
443 000056 002405 BLT 20$; If LT, alpha
444 000060 001402 BEQ 10$; If EQ, dollar
445
446 000062 062700 000011 ADD #22-11,R0
447 000066 062700 177711 ADD #11-100,R0
448 000072 062700 000040 ADD #100-40,R0
449 000102 062700 000040 ADD #40,R0
450 RETURN

```

```

912 .SBTTL Flush events from mailbox
913
914 + **--FLSHMB-Flush events from mailbox
915
916 Flush events from the mailbox.
917
918 -
919 Inputs:
920 R0 = LUN of events to be flushed or zero if all events
921 R3 = Address of mailbox
922 R5 = Address of database descriptor
923
924 Registers modified:
925 R1, R2, R3
926
927 .PSECT
928
929 FLSHMB::SAVRG <R4> ; Get a free register
930 SAVMAP ; Save mapping
931 MOV R3,$MAIBX ; Set up mailbox address
932 ADD #M.MAIL,R3 ; Point to mailbox listhead
933
934 10$: MOV (R3),R4 ; Get next entry on mailbox
935 BEQ 40$; If EQ, no more
936 TST R0 ; Are we flushing the entire list?
937 BEQ 30$; If EQ, yes
938 TSTB C,FNC(R4) ; Is this a connect-type function?
939 BMI 20$; If MI, yes ... no LUN to match
940 CMBP R0,C.MOD(R4) ; Is request for this LUN?
941 BEQ 30$; If EQ, yes
942
943 20$: MOV R4,R3 ; Move on to next entry in list
944 BR 10$; and try again
945
946 30$: MOV (R4),(R3) ; Unlink entry from list
947 MOV C,FNC(R4),R1 ; Get CCB function code
948 BIC #*C<177>,R1 ; Isolate function code
949
950 SAVRG <R0,R3>
951 ASL R1 ; Form word index
952 RECMAP ; Allow dispatch table to be in APR 6
953 CALL @FDISP-2(R1) ; Dispatch to processing routine
954 RESRG <R3,R0>
955 BR 10$; Check next entry
956
957 40$: RESMAP ; Restore mapping
958 RESRG <R4> ; Restore register
959 RETURN
960
961 .PSECT $HIGH
962
963 .CCBRT: CALLR @CCBRT ; Need one more indirection to handle
964 ; vectored exec
965
966 +
967 Resource return dispatch table
968 -
969
970 FDISP: .WORD FLCON ; 1 - connect request
971
972 000340 000356'

```

```

1372 .SBTTL Remove an LLT
1373
1374 +
1375 ***-REMLNK-Remove an LLT
1376 Remove the databases describing a logical link.
1377
1378 Inputs:
1379 R5 = Address of database descriptor
1380
1381 Outputs:
1382 'C' always set
1383
1384 Registers modified:
1385 R0, R1, R3, R4
1386
1387 REMLNK::MAPLLT ; Recover mapping to LLT
1388 MOV N$LLT(R5),R3 ; Recover address of LLT
1389 MOV L$RTQ(R3),R4 ; Get address of message awaiting retransmission
1390 BEQ 10$; If EQ, none
1391 CALL RETRES ; Return resources
1392
1393 10$: .IF DF N$SLI
1394
1395 MOV L$ACC(R3),R4 ; Is there a pending connect/accept CCB?
1396 BEQ 20$; If EQ, no
1397 MOV #S.EABL,C.STS(R4)
1398 MOVB #FC.XCP,C.FNC(R4)
1399 CALL @LLCRS ; Complete the request in error
1400
1401 20$: .ENDC
1402
1403 MOV L$MASQ(R3),R1 ; Get address of message reassembly queue
1404 CALLE FL$SLST ; Flush the list
1405
1406 30$: MOV L$ILSQ(R3),R4 ; Get next I/LS message awaiting ACK
1407 BEQ 40$; If EQ, none
1408 MOV (R4),L$ILSQ(R3) ; Remove message from queue
1409 CALL RETRES ; Return the resources
1410 BR 30$; and try again
1411
1412 40$: CLR R0 ; Get local link address
1413 BISB L$LLA(R3),R0 ;
1414 ASL R0 ; Form word offset
1415 ADD N$LVC+2(R5),R0 ; Compute address in LLT table
1416 CLR (R0) ; Remove entry from the LLT table
1417
1418 CALL DEADB ; Deallocate the ECL database
1419
1420 MOV N$PLLT(R5),R0 ; Get physical address of LLT
1421 MOV #L$LNG,R1 ; Size of the block
1422
1423 .IF DF N$MMLL
1424
1425 CALLX $DEACX,AUX ; Deallocate the block
1426
1427 .IFF
1428

```

|                  |                 |                  |                |                    |
|------------------|-----------------|------------------|----------------|--------------------|
| S.EACR= 100336   | TRMLNK 003450RG | T\$NOPL 000006   | VS.NPV= 000001 | W.SNDQ 000016      |
| S.ECBE= 100204   | TRMNET 003454RG | T\$NRNI 000042   | VS.PRV= 000002 | W.STAT 000002      |
| S.EDBO= 100206   | TRMUSR 003532RG | T\$NRPL 000005   | VZ.NVD= 000000 | W.TMP 000010       |
| S.EERR= 100210   | T\$FLAG 000044  | T\$NRUL 000007   | V\$CTR= 001000 | W.WBL 000026       |
| S.EIDM= 100214   | T\$LIF 000013   | T\$NVR 000001    | WK.ACK= 000001 | X\$SDBT= 000000    |
| S.EINF= 100212   | T\$LIFL 000013  | T\$RPRI 000040   | WK.AST= 000200 | ZTIM2 = ***** GX   |
| S.EIOF= 100373   | T\$LIFO 000013  | T\$SSVC 000034   | WK.DIS= 000010 | \$ALOCX= ***** GX  |
| S.ELNS= 100365   | T\$LIFS 000013  | T\$T5 000030     | WK.INT= 000020 | \$BYTE = ***** GX  |
| S.ELST= 100216   | T\$LIN 000000   | T\$T6 000032     | WK.RCV= 000004 | \$CALLX= ***** GX  |
| S.ELWS= 100220   | T\$LIPS 000006  | T\$KMG= 000000   | WK.SND= 000002 | \$DEACX= ***** GX  |
| S.EMTL= 100222   | T\$LLD 000012   | T\$MIN= 000000   | WORD1 = 000302 | \$ENCD= ***** GX   |
| S.ENOF= 100224   | T\$LLDC 000045  | T.PCB = ***** GX | WORD2 = 000010 | \$IOPKT= ***** GX  |
| S.ENRO= 100332   | T\$LLDL 000012  | T.RCVL= ***** GX | WS.DIP= 000010 | \$LADDR= ***** GX  |
| S.ENSL= 100327   | T\$LLDO 000012  | UISAR6= ***** GX | WS.INT= 000002 | \$MAIBX= ***** GX  |
| S.ENUR= 100331   | T\$LLDS 000012  | USRCNF= ***** GX | WS.KAS= 000004 | \$MENU = ***** GX  |
| S.ECTB= 100372   | T\$LLEN 000046  | USRDIS= ***** GX | WS.STA= 000001 | \$OPDAT= ***** GX  |
| S.ERBO= 100226   | T\$LOPR 000002  | USRDSC= ***** GX | W.CINT 000022  | \$OPLNG= ***** GX  |
| S.ERES= 100377   | T\$LTCL 000024  | USTATE 003516RG  | W.CSND 000020  | \$RCCB = ***** GX  |
| S.ERNS= 100375   | T\$LTIM 000026  | US\$CNF= 000002  | W.CTL 000000   | \$RQNAM= ***** GX  |
| S.ETMI= 100230   | T\$LTPR 000014  | US\$DIS= 000006  | W.KAST 000014  | \$SEGMT= ***** GX  |
| S.EUNN= 100376   | T\$LTPS 000020  | US\$DON= 000000  | W.LLT 000004   | \$SESDB= ***** GX  |
| S.EURO= 100374   | T\$NAPL 000004  | US\$DSC= 000004  | W.LUN 000003   | \$SSHFT= 000001    |
| S.SEOM= 000003   | T\$NFE 000000   | US\$EAC= 000012  | W.MBOX 000012  | \$\$\$ = 000062    |
| S.SSUC= 000001   | T\$NLEN 000010  | US\$WDS= 000010  | W.RCVQ 000024  | .CCBRT 000334R 002 |
| TKTCB = ***** GX | T\$NNUL 000002  | VE.FAI= 177777   | W.SEGZ 000006  | .\$\$\$\$= 000034  |
| TLACHK 003332RG  |                 |                  |                |                    |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
003562 001 (RW,I,LCL,REL,CON)  
\$HIGH 000500 002 (RW,I,LCL,REL,CON)  
Errors detected: 0

### \*\*\* Assembler statistics

Work file reads: 189  
Work file writes: 170  
Size of work file: 36286 Words ( 142 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:01:05.40  
SY:SESSUB11S.V2,[131,134]SESSUB11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCS/PA:1,[131,10]V2,SESSUB

```

46 .IF NDF R$$MPL
47 .IF DF N$$MCP
48 .SBTTL Copy and install TCB
49
50 **-$TCBCP-Copy and install TCB
51
52 This routine is called to copy an installed TCB into an allocated
53 TCB and link it into the system task directory. Specifically,
54 this routine does the following:
55
56 1. Copy the contents of the installed TCB into the allocated TCB
57 2. Initialise all listheads in the allocated TCB.
58 3. Transfer any send packets destined for the allocated TCB
59 4. Set up the task name in the new TCB
60 5. Link the new TCB into the system task directory immediately
61 after the TCB input as the prototype installed TCB
62
63 Inputs:
64 R0 = Address of the allocated TCB
65 R1 = Address of the installed (prototype) TCB
66 R3 = Address of the task name double-word for the allocated TCB
67
68 Outputs:
69 R0 and R1 preserved
70
71 .PSECT $HIGH
72
73 $TCBCP::SAVNR ; Save R4 and R5
74 SAVRG <R0,R1> ; Save TCB addresses
75 MOV R0,R4 ; Copy allocated TCB address
76 MOV TLGTH,R2 ; Get length of TCB
77 ASR R2 ; Set number of words to zero
78 CLR (R4)+ ; Clear all of TCB
79 SOB R2,5$;
80 TST (R0)+ ;
81 MOV T.DPRI(R1),(R0)+ ; Skip over utility link word (T.LNK)
82 CLRB (R0)+ ; Init running priority (T.PRI)
83 CLR (R0)+ ; Init I/O count (T.IOC)
84 MOV (R3)+,(R0)+ ; Point to task name field (T.CPCB)
85 MOV (R3)+,(R0)+ ; Copy task name
86 ADD #T.RCVL,R1 ; Point to installed TCB RCVL list
87 CALL 40$; Setup receive queue
88
89 10$: CALL a(SP)+ ; Mark T.RCVL call
90 CLR (R0)+ ; Setup AST queue
91 CLR (R0)+ ; Clear EFN words (T.EFLG)
92 CMP (R0)+,(R0)+ ;
93 MOV #TS.EXE!TS.OUT,(R0)+ ; Point to status words (T.UCB) (T.TCBL)
94 ADD #T.ST2-T.EFLG,R1 ; Init first status word
95 MOV (R1)+,(R0)+ ; Point to installed TCB second status word
96 BIC #CT2.CHK,(R0)+ ; Copy second status word (T.ST2)
97 MOV (R1)+,(R0)+ ; Clear all but checkpointability (T.ST2)
98 BIC #T3.REMIT3.MCR!T3.CAL,(R0)+ ; Copy third status word (T.ST3)
99 MOV (R1)+,(R0)+ ; Clear necessary bits (T.ST3)
100 MOV (R1)+,(R0)+ ; (T.DPRI+T.LBN)
101 MOV (R1)+,(R0)+ ; (T.LBN+1)
102 MOV (R1)+,(R0)+ ; (T.LDV)

```

|                  |                 |                 |                 |                   |
|------------------|-----------------|-----------------|-----------------|-------------------|
| ACCLLT= ***** GX | D\$BUG= 177514  | LS.MAK= 000020  | L.TMRI 000056   | N\$LLT 000026     |
| ACKCI = ***** GX | D\$ISK= 000000  | LS.MNK= 000040  | L.TYP 000001    | N\$LLTM 000024    |
| A\$CHK= 000000   | D\$LL1= 000001  | LS.RES= 000360  | L.USA 000024    | N\$LCV 000036     |
| A\$CPS= 000000   | D\$YNC= 000000  | LS.RSV= 000300  | L.USTA 000036   | N\$MBXQ 000050    |
| A\$PRI= 000000   | D\$YNM= 000000  | LT.CCA= 000020  | L.VER 000015    | N\$PLLT 000030    |
| A\$TRP= 000000   | ER\$ABM= 000010 | LT.DIR= 000010  | L.WIND 000040   | N\$SLA 000016     |
| BRKLNK= ***** GX | ER\$ABO= 000046 | LT.LCL= 000001  | MA.CI = 000040  | N\$SNOD 000012    |
| CL\$MFL= 000010  | ER\$ABT= 000011 | LT.LPL= 000002  | MA.DA = 000000  | N\$TIM 000004     |
| CL\$SFL= 000004  | ER\$ACC= 000042 | LT.NOT= 000040  | MA.LL = 000020  | N\$VCB 000010     |
| CL\$TYP= 000001  | ER\$CDI= 000052 | LT.RSU= 000200  | MC.CC = 000040  | N\$SACC= 000001   |
| CL.MU1= 000001   | ER\$COM= 000047 | LT.SLI= 000004  | MC.CI = 000020  | N\$SEVL= 000001   |
| CL.MU2= 000002   | ER\$FMT= 000005 | LT.TDA= 000100  | MC.DC = 000100  | N\$SLDV= 000001   |
| CL.RES= 177774   | ER\$MLB= 000006 | L\$ASG= 000000  | MC.DI = 000060  | N\$SMLL= 000001   |
| CV\$MSK= 000003  | ER\$NNF= 000012 | L\$SDRV= 000000 | MC.NO = 000000  | N\$SMOV= 000010   |
| CV\$31 = 000001  | ER\$NOD= 000002 | L\$SP11= 000001 | MC.RC = 000140  | N\$SNCT= 000001   |
| CV\$32 = 000000  | ER\$NSL= 000013 | L\$S11R= 000000 | MD.BM = 000040  | N\$SPEM= 000001   |
| CV\$40 = 000002  | ER\$NSR= 000003 | L.CSTA 000037   | MD.EM = 000100  | N\$SSES= 000001   |
| C\$ORE= 000400   | ER\$RES= 000001 | L.CTR 000074    | MD.LLS= 000040  | P\$P45= 000000    |
| C\$RSH= 177564   | ER\$STA= 000051 | L.DCR 000100    | MD.IM = 000020  | P\$WRD= 000000    |
| DECT = ***** GX  | ER\$UOB= 000004 | L.FLAG 000014   | MF.ACK= 000004  | Q\$OPT= 000010    |
| D\$AMXC 000072   | E\$XPR= 000000  | L.LLSQ 000052   | MF.CTL= 000010  | RTRNS= ***** GX   |
| D\$AMXH 000074   | F\$SLVL= 000001 | L.LLTT 000066   | MF.DAT= 000000  | R\$DER= 000000    |
| D\$ANN 000000    | G\$STTP= 000000 | L.LDA 000032    | M\$CRB= 000124  | R\$K11= 000001    |
| D\$BRPR 000102   | G\$TSS= 000000  | L.LIA 000034    | M\$CRX= 000000  | R\$SND= 000000    |
| D\$BRTM 000100   | G\$STTK= 000000 | L.LLA 000002    | M\$FCS= 000000  | R\$11M= 000000    |
| D\$DELF 000045   | G\$WRD= 000000  | L.LNG 000124    | M\$MGE= 000000  | R\$11S= 000000    |
| D\$DELW 000046   | IN.DAT= 000400  | L.LNO 000026    | M\$MUP= 000000  | SESTIM 000000RG   |
| D\$END = 000104  | IN.ILS= 000001  | L.LPT 000065    | M\$NET= 000000  | ST\$CC = 000004   |
| D\$FNB 000034    | I\$RAR= 000000  | L.LSA 000030    | M\$OVR= 000000  | ST\$CIR= 000006   |
| D\$HIOR 000024   | I\$RDN= 000000  | L.LSFD 000046   | NC.FM0= 000000  | ST\$CIS= 000002   |
| D\$HOST 000022   | K\$CNT= 177546  | L.LSFI 000044   | NC.FM1= 000001  | ST\$DAT= 000010   |
| D\$INAC 000044   | K\$CSR= 177546  | L.LTT 000062    | NC.FM2= 000002  | ST\$DIP= 000012   |
| D\$INCT 000042   | K\$LDC= 000000  | L.MASQ 000070   | NF\$BLK= 000100 | ST\$PND= 000014   |
| D\$IPL 000051    | K\$TPS= 000074  | L.MAST 000073   | NF\$DMO= 000010 | S\$WRG= 000000    |
| D\$LID 000020    | LA.ACK= 100000  | L.MASZ 000072   | NF\$MOU= 000040 | S\$YSZ= 007600    |
| D\$LNAM 000006   | LA.CRS= 020000  | L.NIN 000020    | NF\$RST= 000002 | TIMCC 000122R     |
| D\$LNUM 000014   | LA.MSK= 170000  | L.NXN 000016    | NF\$SCN= 000020 | TIMCIR 000160R    |
| D\$LST 000047    | LA.NAK= 110000  | L.NXTH 000010   | NF\$SHU= 000004 | TIMCIS 000110R    |
| D\$MAXC 000064   | LA.NMS= 010000  | L.OPD 000103    | NF\$TIM= 000200 | TIMDIP 000176R    |
| D\$MAXH 000066   | LA.RES= 040000  | L.OPDL 000102   | N\$SDON= 000000 | TIMDSP 000074R    |
| D\$MAXV 000070   | LA.WND= 040000  | L.REM 000006    | N\$SDI= 000002  | TIMNOP 000072R    |
| D\$MLL 000040    | LD\$LP = 000000 | L.RFC 000050    | N\$SDC= 000004  | TRMNET= ***** GX  |
| D\$MNOD 000041   | LF.DRD= 000004  | L.RLA 000004    | NSAQ 000000     | TRMUSR= ***** GX  |
| D\$NA 000062     | LF.FRC= 000001  | L.RNO 000022    | NSACTL 000032   | T\$KMG= 000000    |
| D\$NBEA 000056   | LF.HFO= 000010  | L.RTQ 000060    | NSCIR 000034    | T\$MIN= 000000    |
| D\$NBRA 000054   | LF.HMF= 000040  | L.RTYD 000055   | NSDLA 000020    | US\$CNF= 000002   |
| D\$NEND= 000054  | LF.HSF= 000020  | L.RTYI 000057   | NSDLY 000014    | US\$DIS= 000006   |
| D\$NLN 000030    | LF.IRD= 000002  | L.SEC 000064    | N\$LEN 000054   | US\$DON= 000000   |
| D\$NN 000060     | LF.MMF= 000200  | L.SEGZ 000076   | N\$ENC 000042   | US\$DSC= 000004   |
| D\$OUTT 000043   | LF.MSF= 000100  | L.STA 000000    | N\$ERRC 000022  | US\$EAC= 000012   |
| D\$RETF 000050   | LS.DLS= 100000  | L.TC 000042     | N\$FLG 000005   | US\$WDS= 000010   |
| D\$RWN 000002    | LS.FCC= 000004  | L.TIC 000043    | N\$FNC 000006   | V\$CTR= 001000    |
| D\$RTMR 000076   | LS.FCO= 000001  | L.TIPD 000013   | N\$GENQ 000052  | X\$SDBT= 000000   |
| D\$SEG 000036    | LS.FCI= 000002  | L.TIP1 000012   | N\$GTM 000015   | \$CALLX= ***** GX |
| D\$SER 000032    | LS.ILS= 100000  | L.TMRD 000054   | N\$HIGH 000033  | \$S\$HFT= 000001  |
| D\$SQL 000052    |                 |                 |                 |                   |



## SYMBOL CROSS REFERENCE

CREF 04.00

| SY      | VL       | VALUE | REFERENCES                                            |
|---------|----------|-------|-------------------------------------------------------|
| ACPIDL  | 000000   | RG    | #8-124                                                |
| AUXTSK  | 000164   | R     | 8-201 #10-288                                         |
| CB.CCB  | = 000002 |       | 10-316                                                |
| CCBDSP  | 000000   | R     | #7-63 8-151                                           |
| CCBRT   | = *****  | GX    | 10-318                                                |
| CMPOV   | = *****  | GX    | 8-126                                                 |
| CS.LST  | = 040000 |       | 10-322 10-327                                         |
| CX.REM  | = 000020 |       | 11-376 11-438                                         |
| CX.REQ  | = 000002 |       | 9-258 11-437                                          |
| CX.RUI  | = 000040 |       | 11-410                                                |
| CX.SMC  | = 000010 |       | 10-304                                                |
| C.ADD   | 000034   |       | 10-292 10-295                                         |
| C.BID   | 000003   |       | 10-316                                                |
| C.BUF   | 000014   |       | 11-412                                                |
| C.BUF2  | 000024   |       | 12-492                                                |
| C.CNT2  | 000030   |       | 12-506                                                |
| C.FNC   | 000010   |       | 8-148 9-249 11-360                                    |
| C.MOD   | 000011   |       | 8-149 9-258 *10-304 *11-376 11-410 *11-437 11-438     |
| C.NSP   | 000004   |       | *11-442                                               |
| C.PRO   | 000042   |       | *10-306                                               |
| C.STS   | 000012   |       | 10-315 10-327                                         |
| DECPT   | = *****  | GX    | 9-246                                                 |
| DI.NCT  | 000042   |       | 9-247                                                 |
| DI.SABO | = 000046 |       | 9-251                                                 |
| IODUN   | 000466   | RG    | #15-667                                               |
| IODUN1  | 000464   | RG    | 15-661 #15-665                                        |
| IOERR   | 000454   | RG    | #15-660                                               |
| IOFIN   | = *****  | GX    | 15-680                                                |
| IOSUC   | 000460   | RG    | #15-663                                               |
| IS.SUC  | = *****  | GX    | 15-663                                                |
| I.FCN   | = *****  | GX    | 8-178 8-182                                           |
| I.LN2   | = *****  | GX    | 8-189 15-678                                          |
| I.PRM   | = *****  | GX    | *15-679                                               |
| I.TCB   | = *****  | GX    | 8-165                                                 |
| KISAR6  | = *****  | GX    | *8-136 *8-150 *8-191 *8-200 *11-443                   |
| NDISP   | 000352   | R     | 11-363 #11-451                                        |
| NF\$BLK | = 000100 |       | 8-154 8-194                                           |
| NF\$SCN | = 000020 |       | 8-197 9-237                                           |
| NF\$TIM | = 000200 |       | 8-127 8-145 8-161 9-267                               |
| NMCLH   | = *****  | GX    | 10-288 10-312                                         |
| NMCON   | 000370   | R     | 11-451 #12-486                                        |
| NMEVT   | 000376   | R     | 11-456 #12-499                                        |
| NMVFY   | 000406   | R     | 11-459 #12-515                                        |
| N\$FLG  | 000005   |       | *8-127 *8-145 *8-154 *8-161 8-194 8-197 *9-237 *9-267 |
| N\$GENO | 000052   |       | 9-239 9-265                                           |
| N\$GTM  | 000015   |       | *9-244 *9-247                                         |
| N\$TIM  | 000004   |       | 8-129 *8-133                                          |
| N\$SACC | = 000001 |       | 11-458 12-513                                         |
| N\$SEVL | = 000001 |       | #4-2                                                  |
| N\$SEXT | = *****  |       | 8-204                                                 |
| N\$SMCP | = *****  |       | 10-297 11-380 11-395 12-486 12-499 13-520             |
| N\$SESS | = 000001 |       | #6-54                                                 |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE    | REFERENCES |        |        |       |        |       |       |       |      |      |
|---------|----------|------------|--------|--------|-------|--------|-------|-------|-------|------|------|
| AC\$DNT | = 000002 | #6-49      |        |        |       |        |       |       |       |      |      |
| AC\$X25 | = 000001 | #6-49      |        |        |       |        |       |       |       |      |      |
| AE\$CIR | = 000003 | #6-49      |        |        |       |        |       |       |       |      |      |
| AE\$LIN | = 000001 | #6-49      |        |        |       |        |       |       |       |      |      |
| AE\$MOD | = 000004 | #6-49      |        |        |       |        |       |       |       |      |      |
| ALOCB   | = *****  | 8-161      |        |        |       |        |       |       |       |      |      |
| ALOCDB  | = 000274 | 7-109      | #8-157 |        |       |        |       |       |       |      |      |
| BYTE3   | = 000300 | 7-112      | 7-112  | #7-112 | 7-112 | #7-112 | 7-112 | 7-112 | 7-112 |      |      |
| CL\$ASZ | = 010500 | #6-49      |        |        |       |        |       |       |       |      |      |
| CL\$DLL | = 000500 | #6-49      | 6-49   | 6-49   | 6-49  | 6-49   | 6-49  | 6-49  | 6-49  | 6-49 | 6-49 |
| CL\$ECL | = 000300 | #6-49      |        |        |       |        |       |       |       |      |      |
| CL\$LDN | = 010400 | #6-49      |        |        |       |        |       |       |       |      |      |
| CL\$MAN | = 000000 | #6-49      |        |        | 6-49  | 6-49   |       |       |       |      |      |
| CL\$PAZ | = 034100 | #6-49      |        |        | 6-49  | 6-49   |       |       |       |      |      |
| CL\$PLH | = 034000 | #6-49      |        |        | 6-49  | 6-49   | 6-49  |       |       |      |      |
| CL\$PLL | = 000600 | #6-49      |        |        |       |        |       |       |       |      |      |
| CL\$PRT | = 034200 | #6-49      |        |        |       |        |       |       |       |      |      |
| CL\$ROU | = 010000 | #6-49      | 6-49   | 6-49   |       |        |       |       |       |      |      |
| CL\$SES | = 000200 | #6-49      | 6-49   | 6-49   |       |        |       |       |       |      |      |
| CL\$SGE | = 035000 | #6-49      | 6-49   | 6-49   |       |        |       |       |       |      |      |
| CL\$SSE | = 035100 | #6-49      | 6-49   | 6-49   | 6-49  | 6-49   | 6-49  | 6-49  | 6-49  | 6-49 | 6-49 |
| CL\$TRN | = 000400 | #6-49      | 6-49   | 6-49   | 6-49  | 6-49   | 6-49  | 6-49  | 6-49  | 6-49 | 6-49 |
| CL\$XL2 | = 013700 | #6-49      |        |        |       |        |       |       |       |      |      |
| CL\$XL3 | = 013600 | #6-49      | 6-49   |        |       |        |       |       |       |      |      |
| CL\$X2S | = 013500 | #6-49      | 6-49   |        |       |        |       |       |       |      |      |
| DEACB   | = *****  | 9-196      |        |        |       |        |       |       |       |      |      |
| DEALDB  | = 000362 | 8-174      | #9-190 |        |       |        |       |       |       |      |      |
| DECPT   | = *****  | 7-127      |        |        |       |        |       |       |       |      |      |
| DEVHD   | = *****  | 7-86       |        |        |       |        |       |       |       |      |      |
| DL\$AST | = 000002 | #6-49      |        |        |       |        |       |       |       |      |      |
| DL\$HLT | = 000000 | #6-49      |        |        |       |        |       |       |       |      |      |
| DL\$IST | = 000001 | #6-49      |        |        |       |        |       |       |       |      |      |
| DL\$MAI | = 000004 | #6-49      |        |        |       |        |       |       |       |      |      |
| DL\$OFF | = 000001 | #6-49      |        |        |       |        |       |       |       |      |      |
| DL\$ON  | = 000000 | #6-49      |        |        |       |        |       |       |       |      |      |
| DL\$RUN | = 000003 | #6-49      |        |        |       |        |       |       |       |      |      |
| DL\$SHU | = 000002 | #6-49      |        |        |       |        |       |       |       |      |      |
| DL\$SYN | = 000005 | #6-49      |        |        |       |        |       |       |       |      |      |
| D\$SEG  | = 000036 | 7-128      | 7-133  | *7-136 |       |        |       |       |       |      |      |
| D\$NAM  | = *****  | 7-89       |        |        |       |        |       |       |       |      |      |
| D\$UCB  | = *****  | 7-94       |        |        |       |        |       |       |       |      |      |
| D\$UNIT | = *****  | 7-91       |        |        |       |        |       |       |       |      |      |
| EF\$ACT | = 000001 | #6-49      |        |        |       |        |       |       |       |      |      |
| EVL\$ES | = *****  | 7-112      |        |        |       |        |       |       |       |      |      |
| EV\$ACF | = 000201 | #6-49      |        |        |       |        |       |       |       |      |      |
| EV\$ADR | = 000420 | #6-49      |        |        |       |        |       |       |       |      |      |
| EV\$ADU | = 000417 | #6-49      |        |        |       |        |       |       |       |      |      |
| EV\$APL | = 000400 | #6-49      |        |        |       |        |       |       |       |      |      |
| EV\$ARC | = 000421 | #6-49      |        |        |       |        |       |       |       |      |      |

```

66 .SBTTL Session control main logic
67 ;+
68 **-$SESON-Session control main logic
69 ;
70 This routine will be entered when the network ACP (session control)
71 is mounted.
72 ;
73 ;
74 000000 .PSECT $HIGH
75
76 000000 $SESON::
77 .IF DF R$$MPL
78 .IF NDF R$$PRO
79 XQT TRNVEC ; Translate executive vector
80 .ENDC
81 .ENDC
82
83 000000 SWSTK$ 10$; Enter system state
84
85 000004 CALL SESINI ;; Initialise session control
86 000010 BCS 100$;; If CS, initialisation failure
87 000012 RETURN
88
89 000014 10$: XQT RCP1 ; Try to start level 1 routing
90 000022 XQT RCP2 ; Try to start level 2 routing
91
92 000030 016705 000000G 20$: MOV $SESDB,R5 ; Point to the ECL database
93 000034 CALL ACPIDL ; Enter main idle loop
94 000040 132765 000010 000005 BITB #NFDMO,N$FLG(R5)
95 000046 001770 BEQ 20$; Loop until we are requested to dismount
96
97 000050 105765 000032 TSTB N$ACTL(R5) ; Are there any active logical links?
98 000054 001365 BNE 20$; If NE, yes ... keep running
99 000056 005765 000050 TST N$MBXQ(R5) ; Are there any active mailboxes?
100 000062 001362 BNE 20$; If NE, yes ... keep running
101
102 000064 SWSTK$ 20$; Enter system state
103 000070 CALLX $XPTDS,XPT ;; Initiate/check for circuit shutdown
104 000100 103004 BCC 30$; If CC, all circuits shutdown
105 000102 152765 000200 000005 BISB #NTIM,NFLG(R5)
106 000110 RETURN ;; Leave system state
107
108 000112 016700 000000G 30$: MOV $UCB,R0 ; Get address of UCB
109 000116 142760 000000G 000000G BICB #US.MDM,U.STS(R0) ; Clear the 'dismount in progress' flag
110 000124 152760 000000G 000000G BISB #US.MNT,U.STS(R0) ; Set the 'not mounted' flag
111 000132 152760 000000G 000000G BISB #US.OFL,U.ST2(R0) ; Set the 'off line' flag
112 000140 005060 000000G CLR U.ACP(R0) ; Dismount the network
113 000144 CALL DEALDB ; Deallocate session control databases
114 000150 CALLX $XPTDC,XPT ; Finalize XPT shutdown
115
116 000160 132765 000004 000005 BITB #N$,N$FLG(R5) ; Did we exit via the shut state
117 000166 001407 BEQ 40$; If EQ, no
118 000170 000170 EVT$ 2,0,,,ROPR,SCOFF+400!SC$SHU
119 000204 000406 BR 50$
120
121 000206 40$: EVT$ 2,0,,REOPR,SCOFF+400!SC$ON
122 000222 142765 000054 000005 50$: BICB #N$MOU,NDMO,NSHU,N$FLG(R5)

```

|     |     |                                          |
|-----|-----|------------------------------------------|
| 6-  | 42  | Macro definitions                        |
| 7-  | 57  | Received message processing              |
| 8-  | 79  | Process received session control message |
| 9-  | 136 | Process received CI message              |
| 10- | 241 | Process a returned CI message            |
| 11- | 275 | Process received CC message              |
| 12- | 351 | Process received DI message              |
| 13- | 405 | Process received DC message              |
| 14- | 450 | Get optional data from message           |
| 15- | 482 | Process requested connect services       |
| 16- | 544 | Send a disconnect initiate message       |
| 17- | 578 | Retransmit disconnect initiate message   |
| 18- | 598 | Retransmit message                       |

```

482 .SBTTL Process requested connect services
483
484 **--PROCON-Process requested connect services
485
486 Process the services requested by the connect initiate or connect
487 confirm message.
488
489 Inputs:
490 R3 = Virtual address of LLT
491 R4 = Address of CCB
492 C.LIN = Channel # message was received on
493 R5 = Address of database descriptor
494
495 Registers modified:
496 R0, R1
497
498 PROCON: CLR R0 ; Assume no type bits to be set
499 MOV N$$SNOD(R5),L.REM(R3)
500 MOV @DECPT,R1 ; Point to our node name
501 CMP N$$SNOD(R5),D$LNUM(R1) ; Check the node source
502 BNE 10$; If NE, connect was not from ourselves
503
504 MOV #LT.LCL*400,R0 ; Assume this is an internal link
505 MOV C.LIN(R4),R1 ; Get the channel #
506 BEQ 10$; If EQ, internal loopback
507 MOV #LT.LPL*400,R0 ; This is a loopback link
508 MOV R1,L.REM(R3) ; Store channel #
509 MOV C.ADD+4(R4),L.NXTH(R3) ; Save next hop
510
511 10$: MOVB $INFO,R1 ; Get remote NSP version number
512 MOV R1,L.VER(R3) ; and save for later
513 BIC #^C<CV$MSK>,R1 ; Isolate remote NSP version #
514 CMP R1,#CV$40 ; Is it version 4.0 or later??
515 BLT 20$; If LT, no
516 BIS #LT.CCA*400,R0 ; Indicate that we can send cross channel ACK's
517
518 20$: BIS R0,(R3) ; Set link type flags
519 MOV N$$SLA(R5),L.RLA(R3)
520 MOVB $SRVCS,R1 ; Get requested services
521 BIT #CL$SFL,R1 ; Segment flow control?
522 BEQ 30$; If EQ, no
523 BISB #LF.HSF,L.FLAG(R3)
524
525 30$: BIT #CL$MFL,R1 ; Message flow control?
526 BEQ 40$; If EQ, no
527 BISB #LF.HMF,L.FLAG(R3)
528
529 40$: MOV @DECPT,R1 ; Point to our node name
530 CMP D$SEG(R1),$SEGMT ; Is requested size larger than our segment size?
531 BHIS 50$; If HS, no
532 MOV D$SEG(R1),$SEGMT ; Use our maximum segment size
533
534 50$: MOV $SEGMT,L.SEGZ(R3)
535
536 .IF DF N$$SLI
537
538 MOVB C.LIN(R4),L.CHN(R3)

```

\*\*FILE\*\*ID\*\*SESQIO

J 6

```
SSSSSSSS EEEEEEEEE SSSSSSS QQQQQQ IIIIII 000000
SSSSSSSS EEEEEEEEE SSSSSSS QQQQQQ IIIIII 000000
SS EE SS QQ QQ II 00 00
SS EE SS QQ QQ II 00 00
SS EE SS QQ QQ II 00 00
SS EE SS QQ QQ II 00 00
SSSSSS EEEEEEEE SSSSSS QQ QQ II 00 00
SSSSSS EEEEEEEE SSSSSS QQ QQ II 00 00
 SS SS QQ QQ II 00 00
 SS SS QQ QQ II 00 00
 SS SS QQ QQ II 00 00
 SS SS QQ QQ II 00 00
SSSSSSS EEEEEEEEE SSSSSSS QQQQ QQ IIIIII 000000
SSSSSSS EEEEEEEEE SSSSSSS QQQQ QQ IIIIII 000000


```

```
 11 11 SSSSSSS
 11 11 SSSSSSS
1111 1111 SS
1111 1111 SS
 11 11 SS
 11 11 SS
 11 11 SSSSSS
 11 11 SSSSSS
 11 11 SS
 11 11 SS
 11 11 SS
 11 11 SS
111111 111111 SSSSSSS
111111 111111 SSSSSSS
```

SESQIO - Session control QIO co MACRO V05.03b Friday 28-Jun-85 19:58<sup>J 7</sup> Page 13-3  
Symbol table

\$\$SHFT= 000001 .\$\$\$\$.= 000034

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000600 001 (RW,I,LCL,REL,CON)

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 102  
Work file writes: 102  
Size of work file: 25949 Words ( 102 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:26.10

SY:SESQIO11S.V2,[131,134]SESQIO11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCS/PA:1,[131,10]V2,SESQIO

```

203 .SBTTL Connect request completion
204
205 ;+
206 **SLICC-Connect request completion
207 A connect request has been completed with a connect confirm (CC)
208 message.
209
210 ; -
211 Inputs:
212 R3 = Virtual address of LLT
213 R5 = Address of database descriptor
214
215 ; Registers modified:
216 R0, R1, R2, R4
217
218 SLICC:: MOV L.ACC(R3),R4 ; Get address of initiating CCB
219 CLR L.ACC(R3) ; None present now
220 MOV L.SEGZ(R3),C.CNT(R4); Return negotiated segment size to user
221
222 MOV #S.SSUC,-(SP) ; Assume successful completion
223 MOV $OPLNG,R0 ; Get length of optional data in message
224 CMP R0,C.CNT2(R4) ; Did user supply enough buffer space?
225 BLOS 10$; If LOS, yes
226 MOV C.CNT2(R4),R0 ; Use user supplied buffer size
227 MOV #S.ELST,(SP) ; Return data lost error
228
229 10$: MOV R0,C.CNT2(R4) ; Return length of optional data
230 BEQ 30$; If EQ, none
231
232 MOV #SOPDAT,R1 ; Point to optional data
233 MAP C.BUF2(R4) ; Map to the supplied buffer
234 MOV C.BUF2+2(R4),R2 ; Point to the supplied buffer
235 20$: MOVB (R1)+,(R2)+ ; Copy the optional data
236 SOB R0,20$; ...
237
238 30$: MOV N$PLLT(R5),C.NSP(R4)
239 MOV (SP)+,R1 ; Get returned status code
240 CALL XMECM2 ; Complete the request
241
242 MOV N$LLT(R5),R3 ; Recover LLT address
243 MAPLLT ; and mapping
244 RETURN

```



```

673 .SBTTL Notify user of logical link disconnect/abort
674 ;+
675 ;*-SLIDIS-Notify user of logical link disconnect/abort
676 ;
677 ; Queue a CCB to the user indicating that the logical link has been
678 ; disconnected or aborted.
679 ;
680 ; Inputs:
681 ; R3 = Virtual address of LLT
682 ; R5 = address of database descriptor
683 ;
684 ; Registers modified:
685 ; R0, R1, R2, R4
686 ;
687 SLIDIS::BIS #LT.DIR*400,(R3); Mark disconnect received from the network
688 ;
689 TST L.TIPI(R3) ; Any transmits outstanding?
690 BNE 100$; If NE, yes ... can't notify user yet
691 ;
692 CALL FLSXMT ; Flush pending transmits
693 CALL @CCBG7 ; Allocate a CCB
694 BCS 100$; If CS, allocation failure
695 ;
696 MOVB #US$WDS,L.USTA(R3)
697 MOV NS$PLLT(R5),C.NSP(R4)
698 MOVB L.ULA(R3),C.FLG2(R4)
699 MOVB L.DCR(R3),C.FLG2+1(R4)
700 NEGB C.FLG2+1(R4) ; Form negative form of disconnect reason
701 MOVB L.PDVC(R3),C.STA(R4)
702 ;
703 CALL PNTCCB ; Point CCB at optional data
704 MOV #S$DSR*400+FC.RCP,C.FNC(R4)
705 CALL @LLCRS ; Queue request to control process
706 ;
707 100$: RETURN

```

|     |      |                                                 |
|-----|------|-------------------------------------------------|
| 6-  | 42   | Macro definitions                               |
| 7-  | 67   | Add entry to general delivery queue             |
| 8-  | 98   | Add a new logical link database                 |
| 9-  | 211  | Add mail to mailbox                             |
| 10- | 243  | Add optional data to message                    |
| 11- | 279  | Allocate an ECL node database                   |
| 12- | 352  | Break the logical link                          |
| 13- | 381  | Convert RAD50 to ASCII                          |
| 14- | 410  | Convert one RAD50 character                     |
| 15- | 451  | CVTAS - Convert address to ASCII                |
| 16- | 511  | Convert ASCII to RAD50                          |
| 17- | 586  | Convert CI message to pending connect block     |
| 18- | 724  | Copy image field                                |
| 19- | 756  | Copy optional outgoing data                     |
| 20- | 789  | Deallocate ECL node database                    |
| 21- | 837  | Flush I/O packets from a window block           |
| 22- | 878  | Flush queue of I/O requests                     |
| 23- | 912  | Flush events from mailbox                       |
| 24- | 977  | Flush pending connect/verification request      |
| 26- | 1048 | Find mailbox                                    |
| 27- | 1078 | Allocate a data buffer for message transmission |
| 28- | 1136 | Requeue I/O request                             |
| 29- | 1159 | Kill logical link                               |
| 30- | 1193 | Map object number to task name                  |
| 31- | 1240 | Point CCB buffer descriptor at optional data    |
| 32- | 1272 | Process image field                             |
| 33- | 1302 | Process descriptor name                         |
| 34- | 1347 | Reject a logical link internally                |
| 35- | 1372 | Remove an LLI                                   |
| 36- | 1437 | Return resources                                |
| 37- | 1469 | Release connect initiate resources              |
| 38- | 1491 | Remove window block resources                   |
| 39- | 1538 | Save optional data in LLI                       |
| 40- | 1562 | Check temporary link address                    |
| 41- | 1620 | Terminate a logical link                        |
| 42- | 1684 | Process user disconnect state                   |

```

451 .SBTTL CVDAS - Convert address to ASCII
452
453 ;+
454 ;**--CVADAS-Convert node address to ASCII and store in connect block
455
456 Inputs:
457 R0 - address of connect block
458 R5 - ECL's ddb address
459
460 Outputs:
461 N.SND in connect block contains ASCII node address
462
463 Registers modified:
464 R1,R5
465
466 CVADAS: SAVRG <R0,R2> ; Save some registers
467 MOV R0,R2 ; Copy address of connect block
468 MOV N$SND(R5),R0 ; Get node address to convert
469 ADD #N.SND,R2 ; Point to name in connect block
470 JSR R4,CVT ; Convert area to ASCII
471 .WORD CVTARE ; ...
472 JSR R4,CVT ; Convert node address to ASCII
473 .WORD CVTADD ; ...
474 RESRG <R2,R0>
475 RETURN
476
477 CVT: SAVRG <R3>
478 MOV (R4)+,R3 ; Get address of conversion table
479 MOV (R3)+,R5 ; Get number of digits to convert
480 10$: MOV (R3)+,-(SP) ; Get divisor
481
482 .IF DF R$EIS
483 MOV R0,R1 ; Set up for divide
484 CLR R0 ; ...
485 DIV (SP)+,R0
486
487 .IFF
488
489 MOV (SP)+,R1
490 CALL @CEDIV
491
492 .ENDC
493
494 ADD #60,R0 ; Convert to ASCII
495 MOVB R0,(R2)+ ; Store ASCII digit
496 MOV R1,R0 ; Get remainder
497 SDB R5,10$
498 RESRG <R3>
499 RTS R4
500
501 CVTARE: .WORD 2
502 .WORD 10240.
503 .WORD 1024.
504
505 CVTADD: .WORD 4
506 .WORD 1000.
507 .WORD 100.

```

|     |        |         |       |        |   |    |                        |
|-----|--------|---------|-------|--------|---|----|------------------------|
| 969 | 000342 | 000000G | .WORD | CMPINT | : | 2  | - interrupt message    |
| 970 | 000344 | 000334' | .WORD | .CCBRT | : | 3  | - user disconnect      |
| 971 | 000346 | 000334' | .WORD | .CCBRT | : | 4  | - user abort           |
| 972 | 000350 | 000334' | .WORD | .CCBRT | : | 5  | - network abort        |
| 973 | 000352 | 003074' | .WORD | RETRES | : | 6  | - network event        |
| 974 | 000354 | 000356' | .WORD | FLVFX  | : | 7  | - network verification |
| 975 |        |         | .WORD | FLMOP  | : | 10 | - mop event            |

J 13

1429

CALL @DEACB ; Deallocate the block

1430

.ENDC

1431

1432

1433

1434

1435

003064 105365 000032  
003070 000261  
003072

DECB N\$ACTL(R5) ; Reduce count of active logical links  
SEC  
RETURN

K 13

| SYMBOL   | VALUE       | REFERENCES |         |         |         |         |      |      |      |
|----------|-------------|------------|---------|---------|---------|---------|------|------|------|
| ACCLLT   | = ***** GX  | 34-1365    | 40-1602 |         |         |         |      |      |      |
| ACSDNT   | = 000002    | #6-54      |         |         |         |         |      |      |      |
| AC\$X25  | = 000001    | #6-54      |         |         |         |         |      |      |      |
| ADDGNQ   | = 000000 RG | #7-80      | 24-1014 |         |         |         |      |      |      |
| ADDLNK   | = 000056 RG | #8-115     |         |         |         |         |      |      |      |
| ADDMAI   | = 000356 RG | #9-224     |         |         |         |         |      |      |      |
| ADDOPT   | = 000426 RG | #10-258    |         |         |         |         |      |      |      |
| AE\$CIR  | = 000003    | #6-54      |         |         |         |         |      |      |      |
| AE\$LIN  | = 000001    | #6-54      |         |         |         |         |      |      |      |
| AE\$MOD  | = 000004    | #6-54      |         |         |         |         |      |      |      |
| ALLDB    | = 000524 RG | 8-167      | #11-297 |         |         |         |      |      |      |
| ALOCB    | = ***** GX  | 17-627     |         |         |         |         |      |      |      |
| BRKLNK   | = 000744 RG | #12-366    | 34-1367 |         |         |         |      |      |      |
| BYTES    | = 000050    | #11-326    | #11-326 | 11-326  | 11-326  | 11-326  |      |      |      |
| CAT5     | = 000774 RG | #16-528    |         |         |         |         |      |      |      |
| CB.SDB   | = 000010    | 36-1450    |         |         |         |         |      |      |      |
| CCBGT    | = ***** GX  | 17-621     |         |         |         |         |      |      |      |
| CCBRT    | = ***** GX  | 17-718     | 23-962  | 36-1464 | 37-1487 |         |      |      |      |
| CEACC    | = ***** GX  | 8-136      | 17-646  | 30-1215 | 38-1527 |         |      |      |      |
| CEDIV    | = ***** GX  | 14-432     | 15-490  |         |         |         |      |      |      |
| CEMUL    | = ***** GX  | 16-565     |         |         |         |         |      |      |      |
| CL\$ASZ  | = 010500    | #6-54      |         |         |         |         |      |      |      |
| CL\$DLL  | = 000500    | #6-54      | 6-54    | 6-54    | 6-54    | 6-54    | 6-54 | 6-54 | 6-54 |
|          |             | 6-54       | 6-54    |         |         |         |      |      |      |
| CL\$ECL  | = 000300    | #6-54      |         |         |         |         |      |      |      |
| CL\$LDN  | = 010400    | #6-54      |         |         |         |         |      |      |      |
| CL\$MAN  | = 000000    | #6-54      |         |         |         |         |      |      |      |
| CL\$PAZ  | = 034100    | #6-54      | 6-54    | 6-54    |         |         |      |      |      |
| CL\$PLH  | = 034000    | #6-54      | 6-54    | 6-54    | 6-54    |         |      |      |      |
| CL\$PLL  | = 000600    | #6-54      |         |         |         |         |      |      |      |
| CL\$PRT  | = 034200    | #6-54      |         |         |         |         |      |      |      |
| CL\$ROU  | = 010000    | #6-54      | 6-54    | 6-54    |         |         |      |      |      |
| CL\$SES  | = 000200    | #6-54      | 6-54    | 6-54    |         |         |      |      |      |
| CL\$SGE  | = 035000    | #6-54      | 6-54    | 6-54    |         |         |      |      |      |
| CL\$SSE  | = 035100    | #6-54      | 6-54    | 6-54    | 6-54    | 6-54    |      |      |      |
| CL\$STRN | = 000400    | #6-54      | 6-54    | 6-54    | 6-54    | 6-54    | 6-54 | 6-54 | 6-54 |
|          |             | 6-54       | 6-54    | 6-54    | 6-54    | 6-54    | 6-54 | 6-54 | 6-54 |
| CL\$XL2  | = 013700    | #6-54      |         |         |         |         |      |      |      |
| CL\$XL3  | = 013600    | #6-54      | 6-54    |         |         |         |      |      |      |
| CL\$X2S  | = 013500    | #6-54      | 6-54    |         |         |         |      |      |      |
| CL.MU1   | = 000001    | 17-680     |         |         |         |         |      |      |      |
| CL.MU2   | = 000002    | 17-699     |         |         |         |         |      |      |      |
| CL.RES   | = 177774    | 17-676     |         |         |         |         |      |      |      |
| CMPINT   | = ***** GX  | 23-969     |         |         |         |         |      |      |      |
| CM.CON   | = 000200    | 40-1608    |         |         |         |         |      |      |      |
| CNTRL    | = 001134 R  | 16-533     | #16-579 |         |         |         |      |      |      |
| CNVCI    | = 001152 RG | #17-608    |         |         |         |         |      |      |      |
| CPYIMG   | = 001662 R  | 17-683     | 17-688  | 17-693  | 17-708  | #18-747 |      |      |      |
| CPYOPT   | = 001702 RG | #19-772    |         |         |         |         |      |      |      |
| CSBGT    | = ***** GX  | 27-1111    |         |         |         |         |      |      |      |
| CSBRT    | = ***** GX  | 36-1467    |         |         |         |         |      |      |      |

```

103 MOV (R1)+,R2 ; Pickup PCB address
104 BIT #PS.SYS,P.STAT(R2) ; System controlled partition?
105 BEQ 15$; If EQ, no
106 MOV P.MAIN(R2),R2 ; Ensure pointing to main partition
107 MOV R2,(R0)+ ; Set PCB pointer (T.PCB)
108 MOV (R1)+,(R0)+ ; (T.MXSZ)
109 CMP (R1)+,(R0)+ ; (T.ACTL)
110 CMP (R1)+,(R0)+ ; (T.SAST)
111 CMP (R1)+,(R0)+ ; (T.CKR, T.TIO)
112 MOV (R1)+,(R0)+ ; Set task size (T.TKSZ)
113
114 BIT #FE.PLA,@FMASK ; Is PLAS available on this system?
115 BEQ 20$; If EQ, no
116 CALL @ (SP)+ ; Set up attachment listhead
117 MOV (R1)+,(R0)+ ; Copy offset word (T.OFF)
118 CLR (R0)+ ; Clear SREF with EFN count (T.SRCT-1)
119 TST (R1)+ ; Advance installed TCB pointer
120 CALL @ (SP)+ ; Set up receive by reference listhead
121 MARK T.RRFL call
122
123 BIT #FE.OFF,@FMASK ; Is parent-offsrng task available on this system
124 BEQ 30$; If EQ, no
125 CALL @ (SP)+ ; Setup DCB listhead
126 CLR (R0) ; Clear rundown count
127
128 MOV FMASK,R4 ; Get address of features mask
129 ADD #2,R4 ; Point to second features word
130 BIT #F2.STP,(R4) ; Is stop-bit synchronization supported ?
131 BEQ 32$; If EQ, no
132 CMP (R1)+,(R0)+ ; (T.EFLM)
133 CMP (R1)+,(R0)+ ; (T.EFLM+2)
134
135 BIT #F2.SWP,(R4) ; Is swapping priority there ?
136 BEQ 33$; If EQ, no
137 CMPB (R0)+,(R1)+ ; (T.SPRI)
138
139 BIT #F2.AHR,(R4) ; Is alternate header placement supported ?
140 BEQ 37$; If EQ, no
141 MOVB (R1)+,(R0)+ ; (T.HDLN) else, copy header length
142
143 TST (SP)+ ; Pop coroutine address
144 RESRG <R1,R0> ; Recover TCB addresses
145 MOV T.TCBL(R1),T.TCBL(R0) ; Link allocated TCB into STD
146 MOV R0,T.TCBL(R1) ; ...
147 RETURN
148
149 CLR (R0) ; Init listhead in allocated TCB
150 MOV R0,2(R0)
151 CMP (SP),#10$; Is it the RCVD list?
152 BEQ 50$; If EQ, yes
153 CMP (SP),#20$; Is it the RRFL list?
154 BNE 80$; If NE, no
155 MOV R1,R5 ; Init pointer to start of list
156 MOV R5,R4 ; Save pointer to previous
157 MOV (R5),R5 ; Point to next in the list
158 BEQ 80$; If EQ, there is none
159 CMP 44(R5),-2(R3) ; Match on first word of task name?

```

J 16  
SESTIM - Session control timer MACRO V05.03b Friday 28-Jun-85 20:01 Page 8-3  
Symbol table

. ABS. 000124 000 (RW,I,GBL,ABS,OVR)  
000224 001 (RW,I,LCL,REL,CON)

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 15706 Words ( 62 Pages)  
Size of core pool: 16552 Words ( 63 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:13.21  
SY:SESTIM11S.V2,[131,134]SESTIM11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCS/PA:1,[131,10]V2,SESTIM



## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                     |
|---------|------------|--------------------------------|
| NSSSI   | = *****    | 7-63                           |
| NSSMC   | = ***** GX | 10-305                         |
| NSSVCT  | = *****    | 8-136                          |
| N.CUIC  | = 000066   | 11-413                         |
| P.HDR   | = ***** GX | 8-167                          |
| WIDDSP  | = 000026 R | #7-94 8-192                    |
| ORMVF   | = ***** GX | 8-158                          |
| RDBRT   | = ***** GX | 9-254 10-320                   |
| REJECT  | = ***** GX | 9-252                          |
| REQTSK  | = 000316 R | 9-260 #11-360                  |
| REQTS1  | = 000334 R | 10-307 #11-365                 |
| RET     | = 000350 R | 8-124 #10-333                  |
| RESMPL  | = *****    | 11-365 11-380                  |
| RESPRO  | = *****    | 11-417                         |
| SCNGNQ  | = 000034 R | 8-137 #9-237                   |
| SESTIM  | = ***** GX | 8-132                          |
| SRTD    | = ***** GX | 11-371                         |
| STPCT   | = ***** GX | 8-220                          |
| TKTCB   | = ***** GX | 8-156                          |
| TSKRT   | = ***** GX | 11-422                         |
| T.PCB   | = ***** GX | 8-166                          |
| T.RCVL  | = ***** GX | 8-157                          |
| T.ST3   | = ***** GX | *11-440                        |
| T3.REM  | = ***** GX | 11-440                         |
| UISAR6  | = ***** GX | 8-136 8-150 8-191 8-200 11-443 |
| VFYNAM  | = ***** GX | 12-515                         |
| XSSHDR  | = *****    | 8-169 15-670 15-682            |
| \$CLQIO | = ***** GX | 7-93                           |
| \$CNQIO | = ***** GX | 7-94                           |
| \$COPT  | = ***** GX | 10-306                         |
| \$CTQIO | = ***** GX | 7-96                           |
| \$DMQ.O | = ***** GX | 7-92                           |
| \$DSQIO | = ***** GX | 7-95                           |
| \$IOPKT | = ***** GX | *8-163 15-667                  |
| \$ROCB  | = ***** GX | *8-147                         |
| \$ROCPY | = ***** GX | *10-305                        |
| \$ROTCB | = ***** GX | *11-378                        |
| \$SESPD | = ***** GX | 8-126                          |
| .SERCP  | = ***** GX | 7-83                           |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE    | REFERENCES |
|---------|----------|------------|
| EV\$AUC | = 000010 | #6-49      |
| EV\$AUS | = 000003 | #6-49      |
| EV\$CDF | = 000520 | #6-49      |
| EV\$COZ | = 000011 | #6-49      |
| EV\$DBR | = 000302 | #6-49      |
| EV\$GAS | = 035101 | #6-49      |
| EV\$HCE | = 035114 | #6-49      |
| EV\$HCI | = 035113 | #6-49      |
| EV\$HFE | = 000506 | #6-49      |
| EV\$IFL | = 000413 | #6-49      |
| EV\$IFO | = 000415 | #6-49      |
| EV\$IFS | = 000414 | #6-49      |
| EV\$INF | = 000515 | #6-49      |
| EV\$LDL | = 000407 | #6-49      |
| EV\$LDN | = 010416 | #6-49      |
| EV\$LDO | = 000411 | #6-49      |
| EV\$LDS | = 000410 | #6-49      |
| EV\$LSC | = 000500 | #6-49      |
| EV\$LUP | = 000412 | #6-49      |
| EV\$NOL | = 000402 | #6-49      |
| EV\$NRC | = 000416 | #6-49      |
| EV\$NSC | = 000200 | #6-49      |
| EV\$NUL | = 000401 | #6-49      |
| EV\$NVR | = 000406 | #6-49      |
| EV\$OPL | = 000403 | #6-49      |
| EV\$PCC | = 034000 | #6-49      |
| EV\$PCI | = 034001 | #6-49      |
| EV\$PCM | = 034002 | #6-49      |
| EV\$PFE | = 000404 | #6-49      |
| EV\$PPC | = 034003 | #6-49      |
| EV\$RCF | = 000517 | #6-49      |
| EV\$RDC | = 010001 | #6-49      |
| EV\$RDR | = 010002 | #6-49      |
| EV\$RJE | = 035106 | #6-49      |
| EV\$RSC | = 000501 | #6-49      |
| EV\$RUL | = 000405 | #6-49      |
| EV\$SNA | = 035000 | #6-49      |
| EV\$SNF | = 000516 | #6-49      |
| EV\$SPE | = 035001 | #6-49      |
| EV\$XCE | = 034110 | #6-49      |
| EV\$XDI | = 013600 | #6-49      |
| EV\$XGW | = 034111 | #6-49      |
| EV\$XMX | = 000514 | #6-49      |
| EV\$XRS | = 000512 | #6-49      |
| EV\$XSC | = 000513 | #6-49      |
| EV\$X2S | = 013500 | #6-49      |
| EV.CCB  | = 000001 | #6-49      |
| EV.CIR  | = 000020 | #6-49      |
| EV.LCB  | = 000100 | #6-49      |
| EV.LIN  | = 000004 | #6-49      |
| EV.MAP  | = 000002 | #6-49      |
| EV.MOD  | = 000040 | #6-49      |

SESMN - Session control main Lo MACRO V05.03b Friday 28-Jun-85 19:57<sup>K 3</sup> Page 8-1  
Session control main logic

```
123 000230 017705 000000G 100$: MOV @TKTCB,R5 ;; Get my TCB address
124 000234 CALLR @DREXT ;; and exit
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

```
.TITLE SESPRO - Session control protocol processing
.IDENT /V05.00/
.ENABL LC
```

```
: Copyright (C) 1982, 1983, 1985 by
: Digital Equipment Corporation, Maynard, MASS.
```

```
: This software is furnished under a license for use only on a
: single computer system and may be copied only with the
: inclusion of the above copyright notice. This software, or
: any other copies thereof, may not be provided or otherwise
: made available to any other person except for use on such
: system and to one who agrees to these license terms. Title
: to and ownership of the software shall at all times remain
: in DEC.
```

```
: The information in this document is subject to change without
: notice and should not be construed as a commitment by Digital
: Equipment Corporation.
```

```
: DEC assumes no responsibility for the use or reliability of
: its software on equipment which is not supplied by DEC.
```

```
: Module description
```

```
: Session control protocol processing
```

```
: Ident history:
```

```
: 4.00 07-NOV-83
: DECNET-11M V4.0
: DECNET-11M-PLUS V2.0
:
: 5.00 22-JUL-85
: DECnet-11M/S V4.2
: DECnet-11M-Plus V3.0
: DECnet-Micro/RSX V1.0
```

SESPRO - Session control protoc MACRO V05.03b Friday 28-Jun-85 19:57<sup>K 5</sup> Page 15-1  
Process requested connect services

539  
540  
541  
542 001550

.ENDC

RETURN

SESPRO - Session control protoc MACRO V05.03b Friday 28-Jun-85 19:57<sup>L 5</sup> Page 16  
Send a disconnect initiate message

SESQIO - Session control QIO co MACRO V05.03b Friday 28-Jun-85 19:58<sup>K6</sup>  
Table of contents

|     |     |                                              |
|-----|-----|----------------------------------------------|
| 6-  | 42  | Macro definitions                            |
| 7-  | 60  | Copy optional data for connect completion    |
| 8-  | 108 | Remove logical link                          |
| 9-  | 130 | Connect request completion                   |
| 10- | 161 | Connect request failure                      |
| 11- | 194 | User disconnect completion                   |
| 12- | 220 | Notify user of logical link disconnect/abort |
| 13- | 269 | Disconnect completion                        |

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                            |
|---------|------------|---------------------------------------|
| ACCLLT  | = ***** GX | 13-293                                |
| ADDMAI  | = ***** GX | 12-262                                |
| CCBGT   | = ***** GX | 12-242                                |
| CCBRT   | = ***** GX | 13-291                                |
| CPYCNC  | 000000 R   | #7-81 9-144 10-181                    |
| C.FNC   | 000010     | *12-254                               |
| C.MOD   | 000011     | *12-260                               |
| C.NSP   | 000004     | *12-247                               |
| C.STS   | 000012     | *12-258 13-290                        |
| DISCMP  | 000520 RG  | #13-283                               |
| EP\$ABT | = 000011   | 12-250                                |
| E\$NBR  | 000014     | #6-49                                 |
| E\$NBS  | 000020     | #6-49                                 |
| E\$NCR  | 000034     | #6-49                                 |
| E\$NCS  | 000036     | #6-49                                 |
| E\$NIC  | 000044     | #6-49                                 |
| E\$NLEN | 000050     | #6-49                                 |
| E\$NLLA | 000012     | #6-49                                 |
| E\$NLNK | 000000     | #6-49                                 |
| E\$NML  | 000040     | #6-49                                 |
| E\$NMR  | 000024     | #6-49                                 |
| E\$NMS  | 000030     | #6-49                                 |
| E\$NNOD | 000002     | #6-49                                 |
| E\$NRT  | 000042     | #6-49                                 |
| E\$NRTP | 000005     | #6-49                                 |
| E\$NSEG | 000010     | #6-49                                 |
| E\$NTIM | 000046     | #6-49                                 |
| E\$NUSE | 000004     | #6-49                                 |
| E\$STRT | 000006     | #6-49                                 |
| FLSHIO  | = ***** GX | 12-265                                |
| IE.DAO  | = ***** GX | 10-184                                |
| IE.NRJ  | = ***** GX | 10-177                                |
| IE.URJ  | = ***** GX | 10-180                                |
| IODUN   | = ***** GX | 9-155 10-190                          |
| IOSUC   | = ***** GX | 11-216 13-298                         |
| IS.DAO  | = ***** GX | 9-147                                 |
| IS.SUC  | = ***** GX | 9-145                                 |
| I.PRM   | = ***** GX | 7-88 7-90 7-97 7-98                   |
| KISAR6  | = ***** GX | *7-97 *7-105 *9-158                   |
| L.CSTA  | 000037     | 11-207                                |
| L.DCR   | 000100     | 10-178 12-247 12-250                  |
| L.SEGZ  | 000076     | 9-151                                 |
| L.USTA  | 000036     | *10-175 *11-213 *12-245 *13-295       |
| L.WIND  | 000040     | 7-84 8-122 9-149 11-210 12-234 12-257 |
| M.USE   | 000010     | *8-125                                |
| NS\$DON | = 000000   | 11-207                                |
| NT.ABO  | = 000005   | 12-252                                |
| NT.ABT  | = 000004   | 12-249                                |
| NT.DSC  | = 000003   | 12-246                                |
| N\$LLT  | 000026     | 9-157                                 |
| N\$LLTM | 000024     | 7-105 9-158                           |
| N\$SBUF | = *****    | 12-237                                |

```

245 .SBTTL Connect request failure
246 ;+
247 ;**--SLICNF--Connect request failure
248 ;
249 ; A connect request failed because the CI message was returned from an
250 ; intermediate node or the CI request timed out.
251 ;--
252 ; Inputs:
253 ; R3 = Virtual address of LLT
254 ; R5 = Address of database descriptor
255 ;
256 ; Registers modified:
257 ; R0, R1, R2, R4
258 ;
259 SLICNF::MOVB #US$DON,L.USTA(R3)
260 MOV L.ACC(R3),R4 ; Get the connect request CCB
261 CLR L.ACC(R3) ; No connect request now
262 MOV L.DCR(R3),R1 ; Get the reason for the failure
263 NEG R1 ; Negate the reason code
264 CALLR XMECM2 ; Complete the request in error

```



709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731

```
.SBTTL Complete connect accept request in error
;+
;**-SLIEAC-Complete connect accept request in error
;
;A disconnect message has been received in response to a connect
;accept message. This is most likely caused by a duplicate connect
;initiate message being received at this node. Return the connect
;accept CCB with an error code.
;-
;Inputs:
;R3 = Virtual address of LLT
;R5 = Address of database descriptor
;
;Registers modified:
;R0, R1, R2, R4
SLIEAC::MOV L,ACC(R3),R4 ; Get the pending connect accept CCB
;CLR L,ACC(R3) ; None present now
;MOV #S.EABS,C.STS(R4)
;MOVB #FC.XCP,C.FNC(R4)
;CALL @LLCRS ; Complete the accept in error
;MOVB #US$DON,L.USTA(R3)
;RETURN
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

```
.TITLE SESSUB - Session control subroutines
.IDENT /V05.00/
.ENABL LC

: Copyright (C) 1982, 1983, 1985 by
: Digital Equipment Corporation, Maynard, MASS.

: This software is furnished under a license for use only on a
: single computer system and may be copied only with the
: inclusion of the above copyright notice. This software, or
: any other copies thereof, may not be provided or otherwise
: made available to any other person except for use on such
: system and to one who agrees to these license terms. Title
: to and ownership of the software shall at all times remain
: in DEC.

: The information in this document is subject to change without
: notice and should not be construed as a commitment by Digital
: Equipment Corporation.

: DEC assumes no responsibility for the use or reliability of
: its software on equipment which is not supplied by DEC.

: Module description

: Session control subroutines

: Ident history:

: 4.00 07-NOV-83
: DECNET-11M V4.0
: DECNET-11M-PLUS V2.0

: 5.00 22-JUL-85
: DECnet-11M/S V4.2
: DECnet-11M-Plus V3.0
: DECnet-Micro/RSX V1.0

:
```

SESSUB - Session control subrou MACRO V05.03b Friday 28-Jun-85 19:59 <sup>K.11</sup> Page 15-1  
CVTDAS - Convert address to ASCII

508 000216 000012  
509 000220 000001

.WORD 10.  
.WORD 1.

SESSUB - Session control subrou MACRO V05.03b Friday 28-Jun-85 19:59 <sup>L.11</sup> Page 16  
Convert ASCII to RAD50

```

977 .SBTTL Flush pending connect/verification request
978
979 *
980 **~FLCON=Flush connect request
981 **~FLVfy=Flush verification request
982
983 Flush a pending connect or verification request from the mailbox
984 queue.
985
986 Inputs:
987 R4 = Address of CCB
988 R5 = Address of database descriptor
989
990 Registers modified:
991 R0, R1, R2, R3
992
993 FLCON:
994 FLVfy: MOV $MAIBX,R0 ; Recover mailbox address
995 DECB M.USE(R0) ; Reduce active link count
996 BITB #CX.GDQ!CX.UNL,C.MOD(R4)
997 BEQ 20$; If EQ, re-queue on general delivery queue
998
999 .IF DF R$$PRO
1000 TST C.PRO(R4) ; Is there a VT: allocated??
1001 BEQ 10$; BR if not, just reject the connect.
1002 CALL REMVT ; Else, dump the virtual terminal.
1003 .IFTF ; DF R$$PRO
1004
1005 1004 000376 012701 000046 10$: MOV #ER$ABO,R1 ; Set reason for rejection
1006 1005 000402 CALLR REJECT ; Reject the connection
1007
1008 1007 000406 20$: .IFT ; DF R$$PRO
1009 1008 TST C.PRO(R4) ; Does this connect have a VT:??
1010 1009 BEQ 10$; BR if not. **TEMP** dont requeue request.
1011 ; ...Later this will cause a new VT to be
1012 ; ...allocated.
1013
1014 1012 .ENDC ; DF R$$PRO
1015 1013 CALLR ADDGNO ; Add CCB to general delivery queue
1016 1014 000406

```

```

1437 .SBTTL Return resources
1438 ;+
1439 ; **--RETRES--Return resources
1440 ; Return the resources described by a CCB.
1441 ;
1442 ; Inputs:
1443 ; R4 = Address of CCB
1444 ;
1445 RETRES::SAVRG <R5> ; Get a free register
1446 003074
1447 1448 003076 116405 000003 MOVB C.BID(R4),R5 ; Get buffer ID field
1449 003102 042705 177761 BIC #^C<16>,R5 ; Isolate code
1450 003106 120527 000010 CMPB R5,#CB.SDB ; Is this a small data buffer?
1451 003112 001003 BNE 10$; If NE, no
1452 003114 016464 000026 000016 MOV C.BUF2+2(R4),C.BUF+2(R4)
1453
1454 003122 017505 003132' 10$: MOV @RETIB-2(R5),R5 ; Get address of routine
1455 003126 CALL (R5) ; Return the buffer
1456
1457 RESRG <R5> ; Recover the register
1458 RETURN
1459
1460 ;+
1461 ; Resource return dispatch table
1462 ;-
1463
1464 003134 000000G RETIB: .WORD CCBRT ; Return a CCB
1465 003136 000000G .WORD LDBRT ; Return a large data buffer
1466 003140 003141' .WORD +1 ; Reserved
1467 003142 000000G .WORD CSBRT ; Return a combined CCB and small data buffer

```

SYMBOL CROSS REFERENCE

CREF 04.00

SYMBOL VALUE REFERENCES

|         |            |          |          |          |          |          |        |        |                 |
|---------|------------|----------|----------|----------|----------|----------|--------|--------|-----------------|
| CTRSES  | = ***** GX | 8-206    | 10-263   | 17-706   |          |          |        |        |                 |
| CVADAS  | 000104 R   | #15-465  | 17-661   |          |          |          |        |        |                 |
| CVT     | 000144 R   | 15-469   | 15-471   | #15-476  |          |          |        |        |                 |
| CVTADD  | 000210 R   | 15-472   | #15-505  |          |          |          |        |        |                 |
| CVTARE  | 000202 R   | 15-470   | #15-501  |          |          |          |        |        |                 |
| CVTBL   | 001144 R   | 16-530   | 16-571   | #16-582  |          |          |        |        |                 |
| CVTC    | 000030 R   | 13-399   | 13-401   | 13-403   | #14-423  |          |        |        |                 |
| CX.GDQ  | = 000001   | 7-80     | 24-995   |          |          |          |        |        |                 |
| CX.REQ  | = 000002   | 7-80     |          |          |          |          |        |        |                 |
| CX.UNL  | = 000004   | 24-995   | 40-1597  |          |          |          |        |        |                 |
| C.BID   | 000003     | 36-1448  |          |          |          |          |        |        |                 |
| C.BUF   | 000014     | 10-266   | *17-634  | *17-642  | 17-664   | 17-670   | 17-713 | 18-749 | 27-1108 27-1116 |
|         |            | *27-1117 | *31-1257 | *31-1265 | *31-1266 | *36-1452 |        |        |                 |
| C.BUF2  | 000024     | *17-635  | 17-640   | *27-1118 | 36-1452  | 37-1484  |        |        |                 |
| C.CNT   | 000020     | *17-698  | *17-704  | *31-1268 |          |          |        |        |                 |
| C.FLG2  | 000032     | *17-633  |          |          |          |          |        |        |                 |
| C.FNC   | 000010     | 23-937   | 23-946   | 40-1608  |          |          |        |        |                 |
| C.LIN   | 000006     | 34-1364  | 40-1601  |          |          |          |        |        |                 |
| C.MOD   | 000011     | *7-80    | 23-939   | 24-995   | *27-1121 | 40-1597  |        |        |                 |
| C.NSP   | 000004     | *27-1122 | 40-1595  |          |          |          |        |        |                 |
| C.RSV   | 000002     | *27-1120 |          |          |          |          |        |        |                 |
| C5TA    | 000000 RG  | #13-399  |          |          |          |          |        |        |                 |
| DEACB   | = ***** GX | 17-716   | 37-1486  | 38-1533  |          |          |        |        |                 |
| DEADB   | 001754 R   | #20-802  | 35-1418  |          |          |          |        |        |                 |
| DECPY   | = ***** GX | 7-92     | 11-334   | 17-613   | 17-643   |          |        |        |                 |
| DL\$AST | = 000002   | #6-54    |          |          |          |          |        |        |                 |
| DL\$HLT | = 000000   | #6-54    |          |          |          |          |        |        |                 |
| DL\$IST | = 000001   | #6-54    |          |          |          |          |        |        |                 |
| DL\$MAI | = 000004   | #6-54    |          |          |          |          |        |        |                 |
| DL\$OFF | = 000001   | #6-54    |          |          |          |          |        |        |                 |
| DL\$ON  | = 000000   | #6-54    |          |          |          |          |        |        |                 |
| DL\$RUN | = 000003   | #6-54    |          |          |          |          |        |        |                 |
| DL\$SHU | = 000002   | #6-54    |          |          |          |          |        |        |                 |
| DL\$SYN | = 000005   | #6-54    |          |          |          |          |        |        |                 |
| DONE    | 003506 R   | #41-1648 | 41-1654  | 41-1662  |          |          |        |        |                 |
| DSTATE  | 000760 R   | 12-367   | #12-374  |          |          |          |        |        |                 |
| D\$INCT | 000042     | 7-93     |          |          |          |          |        |        |                 |
| D\$LNUM | 000014     | 17-614   |          |          |          |          |        |        |                 |
| D\$OUTT | 000043     | 11-335   |          |          |          |          |        |        |                 |
| D\$RNN  | 000002     | 17-644   |          |          |          |          |        |        |                 |
| EF\$ACT | = 000001   | #6-54    |          |          |          |          |        |        |                 |
| ER\$ABO | = 000046   | 24-1004  |          |          |          |          |        |        |                 |
| ER\$FMT | = 000005   | 17-631   |          |          |          |          |        |        |                 |
| ER\$RES | = 000001   | 8-116    | 17-620   | 27-1104  |          |          |        |        |                 |
| EVDSC   | = ***** GX | 11-319   |          |          |          |          |        |        |                 |
| EVLSES  | = ***** GX | 11-326   |          |          |          |          |        |        |                 |
| EV\$ACF | = 000201   | #6-54    |          |          |          |          |        |        |                 |
| EV\$ADR | = 000420   | #6-54    |          |          |          |          |        |        |                 |
| EV\$ADU | = 000417   | #6-54    |          |          |          |          |        |        |                 |
| EV\$APL | = 000400   | #6-54    |          |          |          |          |        |        |                 |
| EV\$ARC | = 000421   | #6-54    |          |          |          |          |        |        |                 |
| EV\$AUC | = 000010   | #6-54    |          |          |          |          |        |        |                 |

```

160 BNE 60$; If NE, no
161 CMP 46(R5),(R3) ; Match on second word of task name?
162 BNE 60$; If NE, no
163 MOV (R5),(R4) ; Unlink packet from list
164 BNE 70$; If NE, was not the last in the list
165 MOV R4,2(R1) ; Update last in list pointer
166 CLR (R5) ; Clear link word
167 MOV R5,@2(R0) ; Link to end of allocated TCB list
168 MOV R5,2(R0) ; Update last in list pointer
169 BR 50$; Restart scan
170
171 80$: CMP (R0)+,(R0)+ ; Bump both pointers
172 CMP (R1)+,(R1)+ ;
173 CALL @ (SP)+ ; Call the caller back
174 BR 40$
175
176 .ENDC
177
178 .ENDC
179
180 000001 .END

```

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES  |
|---------|------------|-------------|
| ACCLLT  | = ***** GX | 7-72        |
| ACKCI   | = ***** GX | 8-150       |
| BRKLNK  | = ***** GX | 8-143       |
| DEOPT   | = ***** GX | 8-130       |
| DSOUTT  | = 000043   | 8-131       |
| ER\$COM | = 000047   | 8-142       |
| LT.LCL  | = 000001   | 8-158       |
| L.LPT   | 000065     | 8-140       |
| L.TMRD  | 000054     | 7-82        |
| NF\$TIM | = 000200   | 7-78        |
| N\$DLY  | 000014     | *8-131      |
| N\$FLG  | 000005     | *7-78       |
| N\$LVC  | 000036     | 7-66 7-67   |
| N\$SEVL | = 000001   | #4-2        |
| N\$SSES | = 000001   | #6-51       |
| N\$SLI  | = *****    | 7-99        |
| N\$SVCT | = *****    | 7-72 8-150  |
| RTRANS  | = ***** GX | 8-138       |
| SESTIM  | 000000 RG  | #7-66       |
| ST\$DAT | = 000010   | 7-76        |
| TIMCC   | 000122 R   | 7-94 #8-132 |
| TIMCIR  | 000160 R   | 7-95 #8-148 |
| TIMCIS  | 000110 R   | 7-93 #8-130 |
| TIMDIP  | 000176 R   | 7-97 #8-157 |
| TIMDSP  | 000074 R   | 7-83 #7-93  |
| TIMNOP  | 000072 R   | #7-87 7-105 |
| TRMNET  | = ***** GX | 8-166       |
| TRMUSR  | = ***** GX | 8-157       |
| \$CALLX | = ***** GX | 7-72 8-150  |



MACRO CROSS REFERENCE CREF 04.00

| MACRO NAME | REFERENCES |        |        |        |        |        |        |       |       |       |
|------------|------------|--------|--------|--------|--------|--------|--------|-------|-------|-------|
| CALL       | 8-124      | 8-132  | 8-137  | 8-151  | 8-158  | 8-192  | 8-201  | 9-252 | 9-254 | 9-260 |
|            | 10-307     | 10-318 | 10-320 | 11-363 | 11-371 | 11-422 | 15-680 |       |       |       |
| CALLR      | 8-220      |        |        |        |        |        |        |       |       |       |
| CCBDF\$    | #6-45      | 6-47   |        |        |        |        |        |       |       |       |
| CNBD\$     | #6-45      | 6-52   |        |        |        |        |        |       |       |       |
| DHBD\$     | #6-45      | 6-48   |        |        |        |        |        |       |       |       |
| ECDD\$     | #6-45      | 6-49   |        |        |        |        |        |       |       |       |
| LLTD\$     | #6-45      | 6-51   |        |        |        |        |        |       |       |       |
| MAP        | #6-44      |        |        |        |        |        |        |       |       |       |
| MAPLLT     | #6-44      |        |        |        |        |        |        |       |       |       |
| MBXDF\$    | #6-45      | 6-50   |        |        |        |        |        |       |       |       |
| RECMAP     | #6-44      | 8-136  | 8-150  | 8-191  | 8-200  | 11-443 |        |       |       |       |
| RESRG      | #6-44      | 11-425 |        |        |        |        |        |       |       |       |
| RETURN     | 9-269      | 10-333 | 11-444 | 12-493 | 12-507 | 12-516 | 15-689 |       |       |       |
| SAVRG      | #6-44      | 11-415 |        |        |        |        |        |       |       |       |
| SWSTK\$    | 8-124      |        |        |        |        |        |        |       |       |       |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE    | REFERENCES |
|----------|----------|------------|
| EV.NOD   | = 000010 | #6-49      |
| EV.PRT   | = 000200 | #6-49      |
| ES\$DATA | 000020   | #6-49      |
| ES\$EVTS | 000000   | #6-49      |
| ES\$LCN  | 000016   | #6-49      |
| ES\$LIN  | 000000   | #6-49      |
| ES\$MOD  | 000012   | #6-49      |
| ES\$NBR  | 000014   | #6-51      |
| ES\$NBS  | 000020   | #6-51      |
| ES\$NCR  | 000034   | #6-51      |
| ES\$NCS  | 000036   | #6-51      |
| ES\$NIC  | 000044   | #6-51      |
| ES\$NLEN | 000050   | #6-51      |
| ES\$NLLA | 000012   | #6-51      |
| ES\$NLNK | 000000   | #6-51      |
| ES\$NML  | 000040   | #6-51      |
| ES\$NMR  | 000024   | #6-51      |
| ES\$NMS  | 000030   | #6-51      |
| ES\$NMOD | 000002   | #6-51      |
| ES\$NOD  | 000010   | #6-49      |
| ES\$NRT  | 000042   | #6-51      |
| ES\$NRTP | 000005   | #6-51      |
| ES\$NSEG | 000010   | #6-51      |
| ES\$NTIM | 000046   | #6-51      |
| ES\$NUSE | 000004   | #6-51      |
| ES\$PORT | 000014   | #6-49      |
| ES\$PRM  | 000002   | #6-49      |
| ES\$STAT | 000006   | #6-49      |
| ES\$STRT | 000006   | #6-51      |
| ES\$TCB  | 000004   | #6-49      |
| E.CTL    | 000020   | #6-49      |
| E.DATA   | 000046   | #6-49      |
| E.EVT    | 000002   | #6-49      |
| E.LCN    | 000042   | #6-49      |
| E.LEN    | 000216   | #6-49      |
| E.LIN    | 000024   | #6-49      |
| E.LNK    | 000000   | #6-49      |
| E.MOD    | 000036   | #6-49      |
| E.NOD    | 000034   | #6-49      |
| E.PDV    | 000021   | #6-49      |
| E.PORT   | 000040   | #6-49      |
| E.PRM    | 000026   | #6-49      |
| E.PVC    | 000044   | #6-49      |
| E.SIZ    | 000022   | #6-49      |
| E.TIME   | 000004   | #6-49      |
| FR\$BCC  | = 000007 | #6-49      |
| FR\$CCF  | = 000001 | #6-49      |
| FR\$CDF  | = 000002 | #6-49      |
| FR\$DAO  | = 000011 | #6-49      |
| FR\$EXC  | = 000000 | #6-49      |
| FR\$FRM  | = 000010 | #6-49      |
| FR\$FTL  | = 000005 | #6-49      |

```

126 .SBTTL Execute a directive
127
128 :+
129 **-$XQTDR-Execute a directive
130 Execute a directive and recover from possible dynamic storage
131 allocation failure.
132
133 : Calling sequence:
134 JSR P5,$XQTDR
135 .WORD <Address of DPB>
136
137 : Outputs:
138 'C' Clear - Directive succeeded
139 'C' Set - Directive failure other than allocation failure
140
141 000240 $XQTDR::DIR$ (R5) ; Execute the directive
142 000244 103011 BCC 20$; If CC, successful
143
144 000246 126727 000000G 000000G CMPB $DSW,#IE.UPN ; Allocation failure?
145 000254 001004 BNE 10$; If NE, no ... return error to caller
146 000256 WSIG$S ; Wait for significant event
147 000264 000765 BR $XQTDR ; Try directive again
148
149 000266 000261 10$: SEC ; Indicate real failure
150 000270 032505 20$: BIT (R5)+,R5 ; Pop return link leaving C-bit alone
151 000272 000205 RTS R5

```

```

42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,CALLE,COUNT$,MAPLLT
45 .MCALL CCBDF$,ECDDBF$,LLTDF$,MSGDF$,CTRDF$,XPTDF$
46 .MCALL DHBDF$
47 000000 CCBDF$; Define CCB offsets
48 000000 ECDDBF$; Define ECL database offsets
49 000000 LLTDF$; Define LLT offsets
50 000000 MSGDF$; Define message values
51 000000 CTRDF$; Define counter block offsets
52 000000 XPTDF$; Define transport message constants
53 000000 DHBDF$; Define DEC home block offsets
54
55 000001 N$$SES = 1 ; This module is part of session control

```

```

544 .SBTTL Send a disconnect initiate message
545
546 **-SENDDI-Send a disconnect initiate message
547
548 Transmit a disconnect initiate message for this logical link.
549
550 Inputs:
551 R3 = Virtual address of LLT
552 R5 = Address of database descriptor
553
554 Registers modified:
555 R0, R1, R2, R4
556
557 001552 SENDDI::CALLE STOPCC ; Ensure stored CC is returned
558 001562 016367 000100 000000G MOV L,DCR(R3), $REASN; Save disconnect reason code
559 001570 004167 000000G JSR R1,GETSDB ; Allocate a small message buffer
560 001574 014 .BYTE NT$DIS ; Disconnect initiate message
561 001575 070 .BYTE MF,CTL!MC.DI
562 001576 103425 BCS 100$; If CS, allocation failure
563
564 001600 116722 000000G MOVB $REASN,(R2)+ ; Fill in disconnect reason
565 001604 116722 000001G MOVB $REASN+1,(R2)+
566
567 001610 CALL ADDOPT ; Add optional user data to message
568
569 001614 CALLE SNSESD ; Transmit the message
570
571 001624 116563 000014 000054 MOVB NSDLY(R5),L.TMRD(R3)
572 001632 017701 000000G MOV @DECP1,R1 ; Get DEC home block address
573 001636 116163 000050 000C55 MOVB DSRETF(R1),L.RTYD(R3) ; Store retransmit factor
574 001644 112763 000004 000037 MOVB #NS$WDC,L.CSTA(R3)
575
576 001652 100$: RETURN

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

```
.TITLE SESQ10 - Session control Q10 completion processing
.IDENT /V05.00/
.ENABL LC
```

```
: Copyright (C) 1982, 1983, 1985 by
: Digital Equipment Corporation, Maynard, MASS.
```

```
: This software is furnished under a license for use only on a
: single computer system and may be copied only with the
: inclusion of the above copyright notice. This software, or
: any other copies thereof, may not be provided or otherwise
: made available to any other person except for use on such
: system and to one who agrees to these license terms. Title
: to and ownership of the software shall at all times remain
: in DEC.
```

```
: The information in this document is subject to change without
: notice and should not be construed as a commitment by Digital
: Equipment Corporation.
```

```
: DEC assumes no responsibility for the use or reliability of
: its software on equipment which is not supplied by DEC.
```

```
: Module description
```

```
: Session control Q10 completion processing
```

```
: Ident history:
```

```
: 4.00 07-NOV-83
: DECnet-11M V4.0
: DECnet-11M-PLUS V2.0
```

```
: 5.00 22-JUL-85
: DECnet-11M/S V4.2
: DECnet-11M-Plus V3.0
: DECnet-Micro/RX V1.0
```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE     | REFERENCES                  |
|---------|-----------|-----------------------------|
| N\$SEVL | = 000001  | #4-2                        |
| N\$MML  | = 000001  | 7-105 9-158                 |
| N\$SES  | = 000001  | #6-58                       |
| N\$VCT  | = *****   | 7-97 7-105 9-158 13-293     |
| PNICCB  | = *****   | 12-255                      |
| RMVLNK  | 000102 R  | #8-121 10-186 11-215 13-297 |
| RMVWND  | = *****   | 8-127                       |
| R\$MPL  | = *****   | 13-283 13-302               |
| T\$FLAG | 000044    | #6-49                       |
| T\$LIF  | 000013    | #5-49                       |
| T\$LIFL | 000013    | #6-49                       |
| T\$LIFO | 000013    | #6-49                       |
| T\$LIFS | 000013    | #6-49                       |
| T\$LIN  | 000000    | #6-49                       |
| T\$LIPS | 000006    | #6-49                       |
| T\$LLD  | 000012    | #6-49                       |
| T\$LLDC | 000045    | #6-49                       |
| T\$LLDL | 000012    | #6-49                       |
| T\$LLDO | 000012    | #6-49                       |
| T\$LLDS | 000012    | #6-49                       |
| T\$LLEN | 000046    | #6-49                       |
| T\$LOPR | 000002    | #6-49                       |
| T\$LTCL | 000024    | #6-49                       |
| T\$LTIM | 000026    | #6-49                       |
| T\$LTFR | 000014    | #6-49                       |
| T\$LTFS | 000020    | #6-49                       |
| T\$NAPL | 000004    | #6-49                       |
| T\$NFE  | 000000    | #6-49                       |
| T\$NLEN | 000010    | #6-49                       |
| T\$NNUL | 000002    | #6-49                       |
| T\$NOPL | 000006    | #6-49                       |
| T\$NRNI | 000042    | #6-49                       |
| T\$NRPL | 000005    | #6-49                       |
| T\$NRUL | 000007    | #6-49                       |
| T\$NVR  | 000001    | #6-49                       |
| T\$RPR  | 000040    | #6-49                       |
| T\$SVC  | 000034    | #6-49                       |
| T\$T5   | 000030    | #6-49                       |
| T\$T6   | 000032    | #6-49                       |
| USRCC   | 000136 RG | #9-144                      |
| USRCNF  | 000220 RG | #10-175                     |
| USRDIS  | 000364 RG | #12-234                     |
| USRDS   | 000312 RG | #11-207                     |
| US\$DON | = 000000  | 10-175 11-213 13-295        |
| US\$WDS | = 000010  | 12-245                      |
| WS.DIP  | = 000010  | 12-259                      |
| W.KAST  | 000014    | 11-211                      |
| W.LLT   | 000004    | 13-292                      |
| W.LUN   | 000003    | 12-260                      |
| W.MBOX  | 000012    | 8-124 12-261                |
| W.SEGZ  | 000006    | *9-151                      |
| W.STAT  | 000002    | *12-259                     |

```

266 .SBTTL SLI accept connection
267 ;+
268 **~SLIACC~SLI accept connection
269 ;-
270 This routine is called to accept an incoming connect request.
271 ;
272 Inputs:
273 R4 = Address of CCB
274 C.NSP - Physical address of LLT (LLA)
275 C.BUF1 - Optional data descriptor
276 C.FLG2 - ULA
277 C.LIN - Flow control options
278 R5 = Address of database descriptor
279 ;
280 SLIACC: MOV C.NSP(R4),R3 ; Get physical address of LLT
281 CALLE ACCLLT ; Gain access to the LLT
282 ;
283 MOVB C.LIN(R4),$FLOW ; Set up requested flow control
284 CLR $LTM ; No long term timer support
285 CALL MOVDAT ; Copy optional data to internal buffer
286 ;
287 MOV #S.SSUC,C.STS(R4)
288 MOV R4,L.ACC(R3) ; Save address of accept CCB
289 MOV L.SEG2(R3),C.CNT(R4)
290 ;
291 CALL SNDACC ; Accept the connection
292 BCC 100$; If CC, wait for ACK response
293 ;
294 CLR L.ACC(R3) ; No pending control function now
295 MOV #S.ERES,R1 ; Resource allocation failure
296 CALL XMECMP ; Complete the request
297 ;
298 100$: RETURN

```



```

733 .SBTTL Flush pending transmits
734
735 + ***-FLSXMT-Flush pending transmits
736 -
737 Flush pending transmits and interrupt transmits for this logical link.
738
739 Inputs:
740 R3 = Virtual address of LLT
741 R5 = Address of database descriptor
742
743 FLSXMT: SAVRG <R4> ; Get a free register
744
745 10$: MOV L.XMTQ(R3),R4 ; Get next transmit CCB
746 BEQ 20$; If EQ, no more
747 MOV (R4),L.XMTQ(R3) ; Unlink CCB from chain
748 CLR (R4) ; Clear link word
749 MOV #S.EERR,C.STS(R4)
750 MOVB #FC.XCP,C.FNC(R4)
751 CALL @LLCRS ; Complete the transmit in error
752 BR 10$; Try again
753
754 20$: MOV L.INTQ(R3),R4 ; Get next interrupt CCB
755 BEQ 30$; If EQ, no more
756 MOV (R4),L.INTQ(R3) ; Unlink CCB from chain
757 CLR (R4) ; Clear link word
758 MOV #S.EERR,C.STS(R4)
759 MOVB #FC.XCP,C.FNC(R4)
760 CALL @LLCRS ; Complete the interrupt transmit in error
761 BR 20$; Try again
762
763 30$: RESRG <R4> ; Restore the register
764 RETURN

```

```

42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,CALLE,EVT$,CALLX,COUNT$
45 .MCALL SAVMAP,RESMAP,MAPLLT,RECMAP
46 .MCALL CCBDF$,ECDDBF$,MBXDF$,CTRDF$,EVLDF$,LLTDF$
47 .MCALL CNBDF$,RNBDF$,LLWDF$,MSGDF$,OBJDF$,NSSYM$
48 .MCALL SLIDF$,DHBDF$
49
50 000000 CCBDF$; Define CCB offsets
51 000000 ECDDBF$; Define ECL database offsets
52 000000 MBXDF$; Define mailbox offsets
53 000000 CTRDF$; Define counter block offsets
54 000000 EVLDF$; Define event logger symbols
55 000000 LLTDF$; Define LLT offsets
56 000000 CNBDF$; Define connect pending block offsets
57 000000 RNBDF$; Define remote name block offsets
58 000000 LLWDF$; Define window block offsets
59 000000 MSGDF$; Define message values
60 000000 OBJDF$; Define object table offsets
61 000000 NSSYM$; Define NS: symbols
62 000000 SLIDF$; Define SLI symbols
63 000000 DHBDF$; Define DEC home block offsets
64
65 000001 N$$SES = 1 ; This module is part of session control

```

```

511 .SBTTL Convert ASCII to RAD50
512
513 **--CAT5-Convert ASCII to RAD50
514
515 Convert a string of up to 3 ASCII characters to RAD50 format.
516
517 Inputs:
518 R0 = Address of next ASCII character
519
520 Outputs:
521 R0 = Address of next ASCII character
522 R1 = Packed RAD50 characters
523 'C' Clear - 3 ASCII characters converted
524 'C' Set - non-RAD50 character detected
525
526 .PSECT
527
528 CAT5:: SAVRG <R3,R4,R5> ; Save some registers
529 CLR R1 ; Initialise result register
530 MOV #CVTBL,R3 ; Get address of conversion table
531
532 10$: MOVB (R0)+,R5 ; Get next character
533 MOV #CNTRL,R4 ; Get address of control string
534 CMPB (R4)+,R5 ; RAD50 character?
535 BLO 60$; If LO, no
536 CMPB (R4)+,R5 ; Alphabetic?
537 BLOS 50$; If LOS, yes
538 CMPB (R4)+,R5 ; RAD50 character?
539 BLO 60$; If LO, no
540 CMPB (R4)+,R5 ; Numeric?
541 BLOS 40$; If LOS, yes
542 CMPB (R4)+,R5 ; '$'?
543 BEQ 30$; If EQ, yes
544 CMPB (R4)+,R5 ; '.'?
545 BEQ 40$; If EQ, yes
546 CMPB (R4)+,R5 ; 'space'?
547 BNE 60$; If NE, no
548
549 C,1052 162705 000027 20$: SUB #-11,R5 ; Space
550 001056 162705 177767 30$: SUB #11-22,R5 ; '$'
551 001062 162705 177722 40$: SUB #22-100,R5 ; Period/digit
552 001066 162705 000100 50$: SUB #100,R5 ; Alphabetic
553
554 .IF DF R$$EIS
555
556 MUL (R3)+,R5 ; Accumulate result
557 ADD R5,R1 ; ...
558
559 .IFF
560
561 001072 SAVRG <R0> ; Save ASCII string pointer
562 001074 010146 MOV R1,-(SP) ; Get result so far
563 001076 010500 MOV R5,R0 ; Get character
564 001100 012301 MOV (R3)+,R1 ; Get next scale factor
565 001102 CALL @CEMUL ; Accumulate result
566 001106 062601 ADD (SP)+,R1 ; ...
567 001110 RESRG <R0> ; Restore ASCII string pointer

```

```

1016 .IF DF R$$PRO
1017 .SBTTL Remove Virtual Terminal
1018
1019 *--REMVT-Remove Virtual Terminal
1020
1021 Dissolve a virtual terminal allocated for this connect.
1022
1023 Inputs:
1024 R4 -> (CB associated with the connect
1025 C.PRO(R4) -> Virtual Terminal UCB
1026
1027 Outputs:
1028 None.
1029
1030 Effects:
1031 If C.PRO contains a pointer to the CO: device UCB, this call is
1032 ignored. If C.PRO points to a virtual terminal UCB, the VT: is deallocated.
1033
1034 REMVT:: SAVRG <R0,R1,R2,R3> ; Save all corruptable registers.
1035 MOV C.PRO(R4),R0 ; Extract pointer to VT: UCB.
1036 BEQ 10$; BR if none.
1037 CMP R0,$COPT ; Is this the CO: UCB??
1038 BEQ 10$; BR is yes, dont try and deallocate it.
1039 MOV #$$DEAVT,-(SP) ; Setup to call $$DEAVT in exec common one.
1040 CALL $MPDC1 ; Deallocate the VT:.
1041 CLR C.PRO(R4) ; Mark the VT: as deallocated.
1042 10$: RESRG <R3,R2,R1,R0> ; Restore the corruptable registers.
1043 RETURN
1044 ;
1045 .ENDC ; DF R$$PRO
1046

```

```

1469 .SBTTL Release connect initiate resources
1470
1471 ;+
1472 ;**--RLSCI-Release connect initiate resources
1473 ;
1474 ; Release the resources allocated for an incoming connect initiate
1475 ; message.
1476 ;
1477 ; Inputs:
1478 ; R4 = Address of CCB
1479 ; R5 = Address of database descriptor
1480 ;
1481 ; Registers modified:
1482 ; R0, R2
1483 RLSCI:: SAVRG <R1,R3> ; Get a some free registers
1484 MOV C.BUF2+2(R4),R0 ; Get pointer to pending connect block
1485 MOV #N.CBL+16.+4,R1 ; and it's size
1486 CALL @DEACB ; Deallocate the block
1487 CALL @CCBRT ; Then release the CCB
1488 RESRG <R3,R1> ; Restore registers
1489 RETURN
1483 003144
1484 003150 016400 000026
1485 003154 012701 000166
1486 003160
1487 003164
1488 003170
1489 003174

```

## SYMBOL CROSS REFERENCE

CREF 04.00

SYMBOL VALUE REFERENCES

|         |          |       |
|---------|----------|-------|
| EV\$AUS | = 000003 | #6-54 |
| EV\$CDF | = 000520 | #6-54 |
| EV\$COZ | = 000011 | #6-54 |
| EV\$DBR | = 000302 | #6-54 |
| EV\$GAS | = 035101 | #6-54 |
| EV\$HCE | = 035114 | #6-54 |
| EV\$HCL | = 035113 | #6-54 |
| EV\$HFE | = 000506 | #6-54 |
| EV\$IFL | = 000413 | #6-54 |
| EV\$IFO | = 000415 | #6-54 |
| EV\$IFS | = 000414 | #6-54 |
| EV\$INF | = 000515 | #6-54 |
| EV\$LDL | = 000407 | #6-54 |
| EV\$LDN | = 010416 | #6-54 |
| EV\$LDO | = 000411 | #6-54 |
| EV\$LDS | = 000410 | #6-54 |
| EV\$LSC | = 000500 | #6-54 |
| EV\$LUP | = 000412 | #6-54 |
| EV\$NOL | = 000402 | #6-54 |
| EV\$NRC | = 000416 | #6-54 |
| EV\$NSC | = 000200 | #6-54 |
| EV\$NUL | = 000401 | #6-54 |
| EV\$NVR | = 000406 | #6-54 |
| EV\$OPL | = 000403 | #6-54 |
| EV\$PCC | = 034000 | #6-54 |
| EV\$PCI | = 034001 | #6-54 |
| EV\$PCM | = 034002 | #6-54 |
| EV\$PFE | = 000404 | #6-54 |
| EV\$PPC | = 034003 | #6-54 |
| EV\$RCF | = 000517 | #6-54 |
| EV\$RDC | = 010001 | #6-54 |
| EV\$RDR | = 010002 | #6-54 |
| EV\$RJE | = 035106 | #6-54 |
| EV\$RSC | = 000501 | #6-54 |
| EV\$RJL | = 000405 | #6-54 |
| EV\$SNA | = 035000 | #6-54 |
| EV\$SNF | = 000516 | #6-54 |
| EV\$SPE | = 035001 | #6-54 |
| EV\$XCE | = 034110 | #6-54 |
| EV\$XDI | = 013600 | #6-54 |
| EV\$XGW | = 034111 | #6-54 |
| EV\$XMW | = 000514 | #6-54 |
| EV\$XRS | = 000512 | #6-54 |
| EV\$XSC | = 000513 | #6-54 |
| EV\$X2S | = 013500 | #6-54 |
| EV.CCB  | = 000001 | #6-54 |
| EV.CIR  | = 000020 | #6-54 |
| EV.LCB  | = 000100 | #6-54 |
| EV.LIN  | = 000004 | #6-54 |
| EV.MAP  | = 000002 | #6-54 |
| EV.MOD  | = 000040 | #6-54 |
| EV.NOD  | = 000010 | #6-54 |

11-326

SESTCB - TCB copy routine for R MACRO V05.03b Friday 28-Jun-85 20:00 L 15 Page 6-3  
Symbol table

|                  |                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|------------------|
| A\$\$CHK= 000000 | F\$\$LVL= 000001 | LD\$LP = 000000  | M\$\$OVR= 000000 | R\$\$DER= 000000 |
| A\$\$CPS= 000000 | G\$\$TPP= 000000 | L\$\$ASG= 000000 | N\$\$ACC= 000001 | R\$\$K11= 000001 |
| A\$\$PRI= 000000 | G\$\$TSS= 000000 | L\$\$DRV= 000000 | N\$\$EVL= 000001 | R\$\$SND= 000000 |
| A\$\$TRP= 000000 | G\$\$TTK= 000000 | L\$\$P17= 000001 | N\$\$LDV= 000001 | R\$\$11M= 000000 |
| C\$\$ORE= 000400 | G\$\$WRD= 000000 | L\$\$11R= 000000 | N\$\$MLL= 000001 | R\$\$11S= 000000 |
| C\$\$RSH= 177564 | I\$\$RAR= 000000 | M\$\$CRB= 000124 | N\$\$MOV= 000010 | S\$\$WRG= 000000 |
| D\$\$BUG= 177514 | I\$\$RDN= 000000 | M\$\$CRX= 000000 | N\$\$NCT= 000001 | S\$\$YSZ= 007600 |
| D\$\$ISK= 000000 | K\$\$CNT= 177546 | M\$\$FCS= 000000 | N\$\$PEM= 000001 | T\$\$KMG= 000000 |
| D\$\$L11= 000001 | K\$\$CSR= 177546 | M\$\$MGE= 000000 | P\$\$P45= 000000 | T\$\$MIN= 000000 |
| D\$\$YNC= 000000 | K\$\$LDC= 000000 | M\$\$MUP= 000000 | P\$\$WRD= 000000 | V\$\$CTR= 001000 |
| D\$\$YNM= 000000 | K\$\$IPS= 000074 | M\$\$NET= 000000 | Q\$\$OPT= 000010 | X\$\$DBT= 000000 |
| E\$\$XPR= 000000 |                  |                  |                  |                  |

. ABS. 000000 000 (RW,I,GBL,ABS,OVR)  
000000 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 8928 Words ( 35 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:06.55  
SY:SESTCB11S.V2,[131,134]SESTCB11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMC/PA:1,[131,10]V2,SESTCB

SESTIM11S CREATED BY MACRO ON 28-JUN-85 AT 20:01 PAGE 2 L 16  
MACRO CROSS REFERENCE CREF 04.00

MACRO NAME REFERENCES

|          |       |       |              |
|----------|-------|-------|--------------|
| CALL     | 7-83  | 8-157 |              |
| CALLE    | #6-44 | 7-72  | 8-150        |
| CALLR    | 8-138 | 8-143 | 8-166        |
| CALLX    | #7-72 | 7-72  | #8-150 8-150 |
| DHDBF\$  | #6-45 | 6-47  |              |
| ECDDBF\$ | #6-45 | 6-48  |              |
| LLTDF\$  | #6-45 | 6-49  |              |
| MAP      | #6-44 |       |              |
| MAPLLT   | #6-44 |       |              |
| RESRG    | #6-44 | 7-84  |              |
| RETURN   | 7-87  | 8-151 |              |
| SAVRG    | #6-44 | 7-80  |              |
| SOB      | 7-86  |       |              |



..FILE..ID..SESIN:

M 1

```
SSSSSSSS EEEEEEEE SSSSSSSS !!!!! NN NN !!!!!
SSSSSSSS EEEEEEEE SSSSSSSS !!!!! NN NN !!!!!
SS EE SS II NN NN II
SS EE SS II NN NN II
SS EE SS II NNNN NN II
SS EE SS II NNNN NN II
SSSSSS EEEEEEEE SSSSSS II NN NN II
SSSSSS EEEEEEEE SSSSSS II NN NN II
SS EE SS II NN NNNN II
SS EE SS II NN NNNN II
SS EE SS II NN NN II
SSSSSSS EEEEEEEE SSSSSSS !!!!! NN NN !!!!!
SSSSSSS EEEEEEEE SSSSSSS !!!!! NN NN !!!!!
```

....
....
....
....

```
11 11 SSSSSSSS
11 11 SSSSSSSS
111 1111 SS
111 1111 SS
11 11 SS
11 11 SS
11 11 SSSSSS
11 11 SSSSSS
11 11 SS
11 11 SS
11 11 SS
11 11 SS
11111 11111 SSSSSSSS
11111 11111 SSSSSSSS
```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE    | REFERENCES                |
|----------|----------|---------------------------|
| FR\$OPN  | = 000004 | #6-49                     |
| FR\$RFD  | = 000006 | #6-49                     |
| FR\$SBU  | = 000012 | #6-49                     |
| FR\$SHO  | = 000003 | #6-49                     |
| FR\$UBU  | = 000013 | #6-49                     |
| FR\$UPT  | = 000014 | #6-49                     |
| ISSAS    | = *****  | 6-48                      |
| MOS\$AC  | = 000016 | #6-49                     |
| MOS\$SPR | = 000012 | #6-49                     |
| MOS\$SSV | = 000014 | #6-49                     |
| MOS\$25A | = 000006 | #6-49                     |
| MOS\$25P | = 000002 | #6-49                     |
| MOS\$25S | = 000004 | #6-49                     |
| MOS\$29S | = 000010 | #6-49                     |
| MS\$HIGH | = 000003 | #6-49                     |
| MS\$100  | = 000000 | #6-49                     |
| MS\$101  | = 000001 | #6-49                     |
| MS\$102  | = 000002 | #6-49                     |
| MS\$103  | = 000003 | #6-49 6-49                |
| NF\$MOU  | = 000040 | 7-121                     |
| NS\$ACTL | 000032   | *8-168                    |
| NS\$FLG  | 000005   | *7-121                    |
| NS\$GENQ | 000052   | *8-170                    |
| NS\$LVC  | 000036   | *8-163 9-190 *9-192 9-193 |
| NS\$MBXQ | 000050   | *8-169                    |
| NS\$VCB  | 000010   | 7-116                     |
| NS\$ECL  | = *****  | 7-112                     |
| NS\$EVL  | = 000001 | #4-2                      |
| NS\$EXT  | = *****  | 7-96                      |
| NS\$GVR  | = 000022 | 7-131                     |
| NS\$SES  | = 000001 | #6-56 7-112               |
| NS\$VCT  | = *****  | 7-103 7-106               |
| OP\$INI  | = 000000 | #6-49                     |
| OP\$TER  | = 000001 | #6-49                     |
| PDVID    | = *****  | 7-75                      |
| PDVTA    | = *****  | 7-79                      |
| PH\$HDE  | = 000004 | #6-49                     |
| PH\$LQC  | = 000002 | #6-49                     |
| PH\$MTS  | = 000003 | #6-49                     |
| PH\$UMP  | = 000000 | #6-49                     |
| PH\$WCS  | = 000001 | #6-49                     |
| RDBSZ    | = *****  | 7-130                     |
| RE\$ADC  | = 000004 | #6-49                     |
| RE\$ADF  | = 000017 | #6-49                     |
| RE\$ADR  | = 000007 | #6-49                     |
| RE\$BLK  | = 000010 | #6-49                     |
| RE\$CAF  | = 000014 | #6-49                     |
| RE\$DAT  | = 000001 | #6-49                     |
| RE\$DRP  | = 000016 | #6-49                     |
| RE\$LDI  | = 000013 | #6-49                     |
| RE\$LSN  | = 000012 | #6-49                     |
| RE\$NML  | = 000001 | #6-49                     |

```

153 .SBTTL Local data areas
154
155 RCP1: RQST$ RCP1.. : Request level 1 routing task
156 RCP2: RQST$ RCP2.. : Request level 2 routing task
157
158 .IF DF R$$$MPL
159 .IF NDF R$$$PRO
160 TRNVEC: GIN$ GI.VEC,$VECTB,$VECLN ; Translate executive vector
161 .ENDC
162 .ENDC
163
164 000000' .END $SESON

```

```

57 .SBTTL Received message processing
58 ;+
59 ;*- .SERCP-Received message processing
60 ; This routine processes a single received session control message.
61 ;
62 ; Inputs:
63 ; R3 = Subfunction code
64 ; R4 = Address of CCB
65 ; R5 = Address of database descriptor
66 ;
67 .PSECT $HIGH
68 000000
69
70 000000 .SERCP::CALLR @RCPTBL(R3) ; Dispatch to processing routine
71
72 ;+
73 ; Receive complete dispatch table
74 ;*-
75
76 000004 000000' RCPTBL: .WORD RCVCTL ; Received control message
77 000006 000000G .WORD USRINT ; Received interrupt message

```

```

578 .SBTTL Retransmit disconnect initiate message
579
580 :+
581 :*-RSNDDI-Retransmit disconnect initiate message
582 :
583 : A timeout has occurred while we are waiting for a DC response to our
584 : disconnect initiate message. Attempt to retransmit the disconnect
585 : initiate message.
586 :
587 : Inputs:
588 : R3 = Virtual address of LLT
589 : R5 = Address of database descriptor
590 :
591 : Registers modified:
592 : R0, R1, R2, R4
593 001654 105363 000055 RSNDDI::DECB L.RTYD(R3) ; Have we exhausted the retry count?
594 001660 003004 BGT RTRANS ; If GT, no ... retransmit the message
595 001662 112763 000000 000037 MOVB #NS$DON,L.CSTA(R3)
596 001670 RETURN

```

```

42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,COUNT$,CALLE,SAVMAP,RESMAP,MAPLLT
45 .MCALL ECDDBS$,CTRDF$,CRBDF$,LLTDF$,MSGDF$,LLWDF$,MBXDF$
46 .MCALL CCBDF$,NSSYMS$
47
48 ECDDBS$; Define ECL database offsets
49 CTRDF$; Define counter block offsets
50 CRBDF$; Define connect request block offsets
51 LLTDF$; Define LLT offsets
52 MSGDF$; Define message values
53 LLWDF$; Define window block offsets
54 MBXDF$; Define mailbox offsets
55 CCBDF$; Define CCB offsets
56 NSSYMS$; Define NS: symbols
57
58 000001 N$$SES = 1 ; This module is part of session control

```

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                                                        |
|---------|------------|-------------------------------------------------------------------|
| W.TMP   | 000010     | 7-85 8-126 9-152 *9-153                                           |
| \$CALLX | = ***** GX | 13-293                                                            |
| \$IOPKT | = ***** GX | *8-121 *8-126 *9-152 10-176 *10-191 11-214 *11-217 13-296 *13-299 |
| \$MAIBX | = ***** GX | *12-261                                                           |
| \$OPDAT | = ***** GX | 7-96                                                              |
| \$OPLNG | = ***** GX | 7-87                                                              |

```

300 .SBTTL SLI connect rejection
301 :+
302 **~SLIREJ~SLI connect rejection
303 :
304 This routine is called to reject an incoming connect request.
305 :
306 Inputs:
307 R4 = Address of CCB
308 C.NSP - Physical address of LLT (LLA)
309 C.BUF1 - Optional data descriptor
310 C.FLG2 - ULA
311 C.LIN - Reject code (negative form)
312 R5 = Address of database descriptor
313 :
314 Registers modified:
315 R0, R1, R2, R3
316 :
317 SLIREJ: MOV C.NSP(R4),R3 ; Get physical address of LLT
318 CALLE ACCLLT ; Gain access to the LLT
319 :
320 CALL MOVDAT ; Move optional data to internal buffer
321 CALL SAVOPT ; and save it in the LLT
322 :
323 MOVB C.LIN(R4),R1 ; Get the disconnect reason
324 NEG R1 ; Convert it for NSP's use
325 CALL BRKLNK ; Break the logical link
326 :
327 MOV #S.SSUC,R1 ; Set up successful completion
328 CALLR XMECMP ; Complete the request

```



```

766 .SBTTL Queue incoming connect to process
767
768 **--QUESLI-Queue incoming connect to process
769
770 Try to queue an incoming connect request to a process.
771
772 Inputs:
773 R3 = Pointer to task/process name (RAD50)
774 R4 = Address of CCB
775 R5 = Address of database descriptor
776
777 Outputs:
778 'C' Clear - Connect successfully queued to process
779 'C' Set - Unable to queue connect
780
781 Registers modified:
782 R0, R1, R2, R3
783
784 QUESLI: MOV (R3)+, R2 ; Get first 3 characters of name
785 TST (R3) ; Do we have more than 3 characters in the name?
786 BNE 100$; If NE, yes ... can't be a process name
787 CALL @PDVID ; Convert process name to PDV index
788 BCS 100$; If CS, no match
789
790 MOV B, C.STA(R4) ; Set up destination process PDV index
791 ADD @PDVIA, R2 ; Get pointer to PDV
792 MOV (R2), R2 ; ...
793 TST Z.PCB(R2) ; Is the process loaded?
794 BEQ 100$; If EQ, no ... can't queue the connect
795 BIT #ZF.SLI, Z.FLG(R2)
796 BEQ 100$; If EQ, process does not support SLI
797
798 SUB #2, C.BUF+2(R4) ; Back up connect block pointer
799 MOV C.BUF+2(R4), R0 ; Get pointer to start of connect block
800
801 .REPT 3
802 MOV N.SND+2(R0), (R0)+
803 .ENDR
804
805 MOV C.CNT(R4), R1 ; Get total size of connect block
806 SUB #N.SFM, R1 ; Compute # of bytes to move
807 ADD #N.SND-N.DFM, R0 ; Get address to start move
808
809 10$: MOV B, N.SFM-N.SND(R0), (R0)+
810 SOB R1, 10$; Remove source node address from connect block
811
812 SUB #4, C.CNT(R4) ; Connect block is 4 bytes shorter
813
814 MOV N$PLLT(R5), C.NSP(R4)
815 MOV #S.SSUC, C.STS(R4)
816 MOV #<S$CNR+400!FC.RCP>, C.FNC(R4)
817 CALL @LLCRS ; Queue incoming request to process
818 MOV N$LLT(R5), R3 ; Recover address of LLT
819 MAPLLT ; Restore mapping to LLT
820 BIS #LT.SLI+400, (R3) ; Mark this as a system level interface link
821
822 TST (PC)+ ; Indicate connect successfully queued

```

```

67 .SBTTL Add entry to general delivery queue
68
69 **--ADDGNQ--Add entry to general delivery queue
70
71 Add an entry to the end of the general delivery queue.
72
73 Inputs:
74 R4 = Address of CCB
75 R5 = Address of database descriptor
76
77 Registers modified:
78 R0, R1
79
80 000000 152764 000003 000011 ADDGNQ::BISB #CX.REQ!CX.GDQ.C.MOD(R4)
81 000006 005014 CLR (R4) ; Clear link word
82 000010 010500 MOV R5,R0 ; Compute address of general delivery queue
83 000012 062700 000052 ADD #N$GENQ,R0 ; Listhead
84
85 000016 010001 10$: MOV R0,R1 ; Save current position in queue
86 000020 011000 MOV (R0),R0 ; Get next queue entry
87 000022 001375 BNE 10$; Scan to the end of the queue
88
89 000024 010411 MOV R4,(R1) ; Link CCB to the end of the queue
90 000026 020465 000052 CMP R4,N$GENQ(R5) ; Is this the first entry in the queue?
91 000032 001005 BNE 20$; If NE, no
92 000034 017700 000000G MOV @DECPT,R0 ; Get address of DEC home block
93 000040 116065 000042 000015 MOVB D$INCT(R0),N$GTM(R5)
94
95 000046 152765 000220 000005 20$: BISB #NF$TIM!NFSCN,NFLG(R5)
96 000054 RETURN

```

```

568
569
570
571 001112 020327 001152'
572 001116 103734
573
574 001120 005727
575 001122 000261
576 001124
577 001132
578
579 001134 132 101 071 CNTRL: .BYTE 'Z','A','9','0','$','.',',
 001137 060 044 056
 001142 040
580
581
582 001144 003100
583 001146 000050
584 001150 000001

 .ENDC

 CMP R3,#CVTBL+6 ; Convert 3 characters yet?
 BLO 10$; If L0, no

 TST (PC)+ ; Indicate successful conversion
 SEC ; Indicate non-RAD50 character encountered
 RESRG <R5,R4,R3> ; Restore registers
 RETURN

 .EVEN

 CVTBL: .WORD 50*50 ; RAD50 pack multipliers
 .WORD 50
 .WORD 1

```

```

1048 .SBTTL Find mailbox
1049
1050 +---FNDMBX-Find mailbox
1051
1052 Find the mailbox associated with a specified task.
1053
1054 -
1055 Inputs:
1056 R0 = Address of TCB to match
1057 R5 = Address of database descriptor
1058
1059 Outputs:
1060 'C' Clear - Mailbox found
1061 'C' Set - Mailbox not found
1062
1063 FNDMBX::SAVRG <R1> ; Get a free register
1064 MOV R5,R1 ; Compute address of mailbox listhead
1065 ADD #N$MBXQ-M.NEXT,R1
1066
1067 10$: MOV M.NEXT(R1),R1 ; Get next mailbox
1068 BEQ 20$; If EQ, no more
1069 CMP R0,M.TASK(R1) ; Is it for this task?
1070 BNE 10$; If NE, no ... keep looking
1071
1072 MOV R1,$MAIBX ; Save address of mailbox
1073 TST (PC)+ ; Indicate mailbox found
1074 SEC ; Indicate mailbox not found
1075 RESRG <R1> ; Recover register
1076 RETURN
1077
1078 000412 010501 000046
1079 000414 062701 000002
1080 000422 016101 000004
1081 000426 001406 000000
1082 000430 020061 000004
1083 000434 001372 000000
1084 000436 010167 000000
1085 000442 005727 000000
1086 000444 000261 000000
1087 000446 000450 000000

```

SESSUB - Session control subrou MACRO V05.03b Friday 28-Jun-85 19:59 Page 38  
Remove window block resources

```

1491 .SBTTL Remove window block resources
1492
1493 **--RMVWND-Remove window block resources
1494
1495 Release the window block associated with a logical link and clean up
1496 the LUT in the task header.
1497
1498 Inputs:
1499 R3 = Address of window block
1500 R5 = Address of database descriptor
1501
1502 Registers modified:
1503 R0, R2, R5
1504
1505 RMVWND::SAVRG <R1>
1506 MOV W.MBOX(R3),R0 ; Get address of mailbox
1507 MOV M.TASK(R0),R0 ; then task's TCB address
1508 MOV I.PCB(R0),R0 ; then PCB address
1509 MOV P.HDR(R0),R1 ; Get address of task's header
1510
1511 .IF DF X$$HDR
1512
1513 BNE 10$; If NE, header is in primary pool
1514 MAP P.REL(R0) ; Map to the header
1515 MOV #140000,R1 ; Set up virtual address of task header
1516
1517 10$:
1518 .ENDC
1519
1520 CLR R2 ; Get LUN used for this logical link
1521 BLSB W.LUN(R3),R2 ;
1522 ASL R2 ; Convert LUN to double word offset
1523 ASL R2 ;
1524 ADD R1,R2 ; Compute address of task header
1525 CLR H.LUN-2(R2) ; No window block now
1526
1527 MOV W.LLT(R3),-(SP) ; Get the window block address
1528 CALL @CEACB ; Map the LLT
1529 MOV (SP)+,R0 ; Get the LLT address
1530 BEQ 20$; If EQ, none
1531 CLR L.WIND(R0) ; Clear the window block address
1532 MOV R3,R0 ; Copy address of window block
1533 MOV #W.WBL,R1 ; Size of a window block
1534 CALL @DEACB ; Deallocate the window block
1535
1536 20$:
1537 MOV R3,R0
1538 CALL @DEACB
1539
1540 RESRG <R1>
1541 RETURN

```

SESSUB - Session control subrou MACRO V05.03b Friday 28-Jun-85 19:59 Page 39  
Save optional data in LLT

SESSUB11S CREATED BY MACRO ON 28-JUN-85 AT 20:00 PAGE 4 M 14  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL | VALUE      | REFERENCES                                 |
|--------|------------|--------------------------------------------|
| EV.PRT | = 000200   | #6-54                                      |
| ESDATA | 000020     | #6-54                                      |
| ESVTS  | 000000     | #6-54                                      |
| ESLCN  | 000016     | #6-54                                      |
| ESLIN  | 000000     | #6-54                                      |
| ESMOD  | 000012     | #6-54                                      |
| ESNBR  | 000014     | #6-53 17-706                               |
| ESNBS  | 000020     | #6-53 10-263                               |
| ESNCR  | 000034     | #6-53                                      |
| ESNCS  | 000036     | #6-53                                      |
| ESNIC  | 000044     | #6-53                                      |
| ESNLEN | 000050     | #6-53 11-326 11-329                        |
| ESNLLA | 000012     | #6-53 20-816 *20-824                       |
| ESNLNK | 000000     | #6-53                                      |
| ESNML  | 000040     | #6-53 11-339 *11-341                       |
| ESNMR  | 000024     | #6-53                                      |
| ESNMS  | 000030     | #6-53                                      |
| ESNNOD | 000002     | #6-53 11-305 11-321 *11-337                |
| ESNOD  | 000010     | #6-54 11-320                               |
| ESNRT  | 000042     | #6-53                                      |
| ESNRT  | 000005     | #6-53 *11-335 11-343                       |
| ESNSEG | 000010     | #6-53                                      |
| ESNTIM | 000046     | #6-53 *11-333                              |
| ESNUSE | 000004     | #6-53 11-307 *11-338 11-339 11-341 *20-820 |
| ESPORT | 000014     | #6-54                                      |
| ESPRM  | 000002     | #6-54                                      |
| ESSTAT | 000006     | #6-54                                      |
| ESSTR  | 000006     | #6-53 *20-818 *20-823                      |
| ESTCB  | 000004     | #6-54                                      |
| CTL    | 000020     | #6-54                                      |
| DATA   | 000046     | #6-54                                      |
| EVT    | 000002     | #6-54                                      |
| LCN    | 000042     | #6-54                                      |
| LEN    | 000216     | #6-54                                      |
| LIN    | 000024     | #6-54                                      |
| LNK    | 000000     | #6-54                                      |
| MOD    | 000036     | #6-54                                      |
| NOD    | 000034     | #6-54                                      |
| PDV    | 000021     | #6-54                                      |
| PORT   | 000040     | #6-54                                      |
| PRM    | 000026     | #6-54                                      |
| PVC    | 000044     | #6-54                                      |
| SIZ    | 000022     | #6-54                                      |
| TIME   | 000004     | #6-54                                      |
| DISP   | 000340 R   | 23-952 #23-968                             |
| LCON   | 000356 R   | 23-968 #24-992                             |
| LSHIO  | 002100 RG  | #21-850                                    |
| LSHMB  | 002124 RG  | #23-928                                    |
| LSHRO  | 000306 R   | 21-867 21-873 #22-892 22-909               |
| LSHT   | 000222 R   | 21-852 #21-857                             |
| LSLST  | = ***** GX | 23-1404                                    |
| LVFY   | 000356 R   | 23-974 #24-993                             |

SESTCB11S CREATED BY MACRO ON 28-JUN-85 AT 20:00 PAGE 1 M 15  
SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL | VALUE    | REFERENCES |
|--------|----------|------------|
| N\$EVL | = 000001 | #4-2       |
| N\$MCP | = *****  | 6-47       |
| R\$MPL | = *****  | 6-46       |

```

SSSSSSSS EEEEEEEEE SSSSSSSS UU UU SSSSSSSS RRRRRRRR
SSSSSSSS EEEEEEEEE SSSSSSSS UU UU SSSSSSSS RRRRRRRR
SS SS EE EE SS SS SS RR RR
SS SS EE EE SS SS SS RR RR
SS SS EE EE SS SS SS RR RR
SSSSSSS EEEEEEEEE SSSSSSSS UU UU SSSSSSSS RRRRRRRR
SSSSSSS EEEEEEEEE SSSSSSSS UU UU SSSSSSSS RRRRRRRR
SS SS EE EE SS SS SS RR RR
SS SS EE EE SS SS SS RR RR
SSSSSSSS EEEEEEEEE SSSSSSSS UUUUUUUUU SSSSSSSS RR RR
SSSSSSSS EEEEEEEEE SSSSSSSS UUUUUUUUU SSSSSSSS RR RR

```

....  
....  
....  
....

```

11 11 SSSSSSSS
11 11 SSSSSSSS
1111 1111 SS
1111 1111 SS
11 11 SS
11 11 SS
11 11 SSSSSS
11 11 SSSSSS
11 11 SS
11 11 SS
11 11 SS
11 11 SS
111111 111111 SSSSSSSS
111111 111111 SSSSSSSS

```



SESIN: - Session control initia MACRO V05.03b Friday 28-Jun-85 19:56<sup>N 1</sup>  
Table of contents

|    |     |                                      |
|----|-----|--------------------------------------|
| 6- | 42  | Macro definitions                    |
| 7- | 58  | Initialise session control ACP       |
| 8- | 142 | Allocate session control databases   |
| 9- | 178 | Deallocate session control databases |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE    | REFERENCES |
|----------|----------|------------|
| RESOPE   | = 000004 | #6-49      |
| RESOPR   | = 000000 | #6-49      |
| RESRCV   | = 000001 | #6-49      |
| RESSED   | = 000011 | #6-49      |
| RESSKW   | = 000006 | #6-49      |
| RESSTA   | = 000002 | #6-49      |
| RESSUM   | = 000003 | #6-49      |
| RESSYN   | = 000000 | #6-49      |
| RESTME   | = 000021 | #6-49      |
| RESTMO   | = 000000 | #6-49      |
| RESTMR   | = 000020 | #6-49      |
| RESUPT   | = 000002 | #6-49      |
| RESURE   | = 000003 | #6-49      |
| RESVER   | = 000005 | #6-49      |
| RESVRQ   | = 000015 | #6-49      |
| RT\$INI  | = 000002 | #6-49      |
| RT\$OFF  | = 000001 | #6-49      |
| RT\$ON   | = 000000 | #6-49      |
| R\$\$11D | = *****  | 6-48       |
| R\$\$11M | = 000000 | 6-48       |
| R\$\$11S | = 000000 | 6-48       |
| SC\$OFF  | = 000001 | #6-49      |
| SC\$ON   | = 000000 | #6-49      |
| SC\$RST  | = 000003 | #6-49      |
| SC\$SHU  | = 000002 | #6-49      |
| SESINI   | = 000000 | #7-74      |
| SES\$DV  | = 051516 | #6-54      |
| SV\$DUM  | = 000001 | #6-49      |
| SV\$LOA  | = 000000 | #6-49      |
| TKTCB    | = *****  | 7-117      |
| TTNS     | = *****  | 7-123      |
| T\$FLAG  | 000044   | #6-51      |
| T\$LIF   | 000013   | #6-51      |
| T\$LIFL  | 000013   | #6-51      |
| T\$LIFO  | 000013   | #6-51      |
| T\$LIFS  | 000013   | #6-51      |
| T\$LIN   | 000000   | #6-51      |
| T\$LIPS  | 000006   | #6-51      |
| T\$LLD   | 000012   | #6-51      |
| T\$LLDC  | 000045   | #6-51      |
| T\$LLDL  | 000012   | #6-51      |
| T\$LLDO  | 000012   | #6-51      |
| T\$LLDS  | 000012   | #6-51      |
| T\$LLEN  | 000046   | #6-51      |
| T\$LCPR  | 000002   | #6-51      |
| T\$LTCL  | 000024   | #6-51      |
| T\$LIIM  | 000026   | #6-51      |
| T\$LTPR  | 000014   | #6-51      |
| T\$LTPS  | 000020   | #6-51      |
| T\$NAPL  | 000004   | #6-51      |
| T\$NFE   | 000000   | #6-51      |
| T\$NLEN  | 000010   | #6-51      |

## Symbol table

|                  |                 |                  |                  |                   |     |
|------------------|-----------------|------------------|------------------|-------------------|-----|
| ACPIDL= ***** GX | EV\$DBR= 000302 | E\$PRM 000002    | M\$SCRB= 000124  | Q\$SOPT= 000010   |     |
| AC\$DNT= 000002  | EV\$GAS= 035101 | E\$STAT 000006   | M\$SCRX= 000000  | RCP1 000274R      | 002 |
| AC\$X25= 000001  | EV\$HCE= 035114 | E\$TCB 000004    | M\$SFCS= 000000  | RCP2 000312R      | 002 |
| AE\$CIR= 000003  | EV\$HC= 035113  | E\$XPR= 000000   | M\$SMGE= 000000  | RESADC= 000004    |     |
| AE\$LIN= 000001  | EV\$HFE= 000506 | E\$CTL 000020    | M\$SMUP= 000000  | RESADF= 000017    |     |
| AE\$MOD= 000004  | EV\$IFI= 000413 | E\$DATA 000046   | M\$SNET= 000000  | RESADR= 000007    |     |
| AE\$CHK= 000000  | EV\$IFO= 000415 | E\$EVT 000002    | M\$SDVR= 000000  | RESBLK= 000010    |     |
| AE\$CPS= 000000  | EV\$IFS= 000414 | E\$LCN 000042    | M\$S100= 000000  | RESCAF= 000014    |     |
| AE\$PRI= 000000  | EV\$INF= 000515 | E\$LEN 000216    | M\$S101= 000001  | RESDAT= 000001    |     |
| AE\$TRP= 000000  | EV\$LDL= 000407 | E\$LIN 000024    | M\$S102= 000002  | RESDRP= 000016    |     |
| BYTE3 = 000300   | EV\$LDN= 010416 | E\$LNK 000000    | M\$S103= 000003  | RESLDT= 000013    |     |
| CL\$ASZ= 010500  | EV\$LDS= 000410 | E\$MOD 000036    | NF\$BLK= 000100  | RESLSN= 000012    |     |
| CL\$DLL= 000500  | EV\$LDS= 000410 | E\$NOD 000034    | NF\$DMO= 000010  | RESNML= 000001    |     |
| CL\$ECL= 000300  | EV\$LSC= 000500 | E\$POV 000021    | NF\$MOU= 000040  | RESOPE= 000004    |     |
| CL\$LDN= 010400  | EV\$LUP= 000412 | E\$PORT 000040   | NF\$RST= 000002  | RESOPR= 000000    |     |
| CL\$MAN= 000000  | EV\$NOL= 000402 | E\$PRM 000026    | NF\$SCN= 000020  | RESRCV= 000001    |     |
| CL\$PAZ= 034100  | EV\$NRC= 000416 | E\$PVC 000044    | NF\$SHU= 000004  | RESSED= 000011    |     |
| CL\$PLH= 034000  | EV\$NSC= 000200 | E\$SIZ 000022    | NF\$STIM= 000200 | RESSKW= 000006    |     |
| CL\$PLL= 000600  | EV\$NUL= 000401 | E\$TIME 000004   | NSACQ 000000     | RESSTA= 000002    |     |
| CL\$PRT= 034200  | EV\$NVR= 000406 | FR\$BCC= 000007  | NSACTL 000032    | RESSUM= 000003    |     |
| CL\$ROU= 010000  | EV\$OPL= 000403 | FR\$CCF= 000001  | NSCIR 000034     | RESSYN= 000000    |     |
| CL\$SES= 000200  | EV\$PCC= 034000 | FR\$CDF= 000002  | NSDLA 000020     | RESSTME= 000021   |     |
| CL\$SGE= 035000  | EV\$PCI= 034001 | FR\$DAD= 000011  | NSDLY 000014     | RESSTMO= 000000   |     |
| CL\$SSE= 035100  | EV\$PCM= 034002 | FR\$EXC= 000000  | NSLEEN 000054    | RESSTMR= 000020   |     |
| CL\$STRN= 000400 | EV\$PFE= 000404 | FR\$FRM= 000010  | NSENC 000042     | RESUPT= 000002    |     |
| CL\$XL2= 013700  | EV\$PPP= 034003 | FR\$FTL= 000005  | NSERRC 000022    | RESURE= 000003    |     |
| CL\$XL3= 013600  | EV\$RCF= 000517 | FR\$OPN= 000004  | NSFLG 000005     | RESVER= 000005    |     |
| CL\$X2S= 013500  | EV\$RDC= 010001 | FR\$RFD= 000006  | NSFNC 000006     | RESVRO= 000015    |     |
| C\$SDFE= 000400  | EV\$RDR= 010002 | FR\$BU= 000012   | NSGENQ 000052    | RT\$INI= 000002   |     |
| C\$SRSH= 177564  | EV\$RJE= 035106 | FR\$SHO= 000003  | NSGTM 000015     | RT\$OFF= 000001   |     |
| DEALDB= ***** GX | EV\$RSC= 000501 | FR\$UBU= 000013  | NSHIGH 000033    | RT\$ON= 000000    |     |
| DL\$AST= 000002  | EV\$RUL= 000405 | FR\$UPT= 000014  | NSLLT 000026     | R\$SDER= 000000   |     |
| DL\$HLT= 000000  | EV\$SNA= 035000 | FR\$LVL= 000001  | NSLLTM 000024    | R\$SK11= 000001   |     |
| DL\$IST= 000001  | EV\$SNF= 000516 | G\$STPP= 000000  | NSLVC 000036     | R\$SND= 000000    |     |
| DL\$MAI= 000004  | EV\$SPE= 035001 | G\$STSS= 000000  | NSMBXQ 000050    | R\$STIM= 000000   |     |
| DL\$OFF= 000001  | EV\$XCE= 034110 | G\$STTK= 000000  | NSPLLT 000030    | R\$STIS= 000000   |     |
| DL\$ON= 000000   | EV\$XDI= 013600 | G\$SWRD= 000000  | NSSLA 000016     | R.QSGC= 000015    |     |
| DL\$RUN= 000003  | EV\$XGW= 034111 | IE.UPN= ***** GX | NSNOD 000012     | R.QSPC= 000014    |     |
| DL\$SHU= 000002  | EV\$XMX= 000514 | I\$RAR= 000000   | NSTIM 000004     | R.QSPN= 000006    |     |
| DL\$SYN= 000005  | EV\$XRS= 000512 | I\$SRDN= 000000  | NSVCB 000010     | R.QSPR= 000012    |     |
| DREXT= ***** GX  | EV\$XSC= 000513 | K\$CNT= 177546   | NSVACC= 000001   | R.QSTN= 000002    |     |
| D\$BUC= 177514   | EV\$X2S= 013500 | K\$CSR= 177546   | NSSEVL= 000001   | SC\$OFF= 000001   |     |
| D\$BIX= 000000   | EV\$CCB= 000001 | K\$SLDC= 000000  | NSSLDV= 000001   | SC\$ON= 000000    |     |
| D\$B11= 000001   | EV\$CIR= 000020 | K\$STPS= 000074  | NS\$MLL= 000001  | SC\$RST= 000003   |     |
| D\$SYNC= 000000  | EV\$LCB= 000100 | LD\$LP= 000000   | NS\$MOV= 000010  | SC\$SHU= 000002   |     |
| D\$SYNM= 000000  | EV\$LIN= 000004 | L\$ASG= 000000   | NS\$NCT= 000001  | SE\$INI= ***** GX |     |
| EF\$ACT= 000001  | EV\$MAP= 000002 | L\$DRV= 000000   | NS\$PEM= 000001  | SV\$DUM= 000001   |     |
| EVLSES= ***** GX | EV\$MOD= 000040 | L\$P11= 000001   | NS\$SES= 000001  | SV\$LOA= 000000   |     |
| EV\$ACF= 000201  | EV\$NOD= 000010 | L\$P11R= 000000  | OP\$INI= 000000  | SV\$WRG= 000000   |     |
| EV\$ADR= 000420  | EV\$PRT= 000200 | MD\$SAC= 000016  | OP\$TER= 000001  | SV\$YSZ= 007600   |     |
| EV\$ADU= 000417  | E\$DATA 000020  | MD\$SPR= 000012  | PH\$HDE= 000004  | TKTCB = ***** GX  |     |
| EV\$APL= 000400  | E\$EVTs 000000  | MD\$SSV= 000014  | PH\$LOC= 000002  | T\$KMG= 000000    |     |
| EV\$ARC= 000421  | E\$LCN 000016   | MD\$25A= 000006  | PH\$MTS= 000003  | T\$MIN= 000000    |     |
| EV\$AUC= 000010  | E\$LIN 000000   | MD\$25P= 000002  | PH\$UMP= 000000  | US.MDM= ***** GX  |     |
| EV\$AUS= 000003  | E\$MOD 000012   | MD\$25S= 000004  | PH\$WCS= 000001  | US.MNT= ***** GX  |     |
| EV\$CDF= 000520  | E\$NOD 000010   | MD\$29S= 000010  | PH\$WCS= 000000  | US.OFL= ***** GX  |     |
| EV\$COZ= 000011  | E\$PORT 000014  | M\$HIGH= 000003  | P\$SAD= 000000   | U.ACP = ***** GX  |     |

```

79 .SBTTL Process received session control message
80
81 ;+
82 **--RCVCTL-Process received session control message
83 Process the received session control message.
84
85 Inputs:
86 R4 = Address of CCB
87 C.BUF+2 - Pointer to start of NSP message
88 C.STS - Pointer to first byte of route header
89 (zero if none)
90 C.FLG2 - Source node address
91 R5 = Address of database descriptor
92
93 Outputs: (to message processing routines)
94 R2 = Pointer to field following source link address
95 R4 = Address of CCB
96 R5 = Address of database descriptor
97 ;+
98 .PSECT
99
100 000000
101 000006
102 000014 016402 000016
103 000020 112201
104 000022 006201
105 000024 006201
106 000026 006201
107 000030 042701 177761
108
109 000034 112265 000020
110 000040 112265 000021
111 000044 112265 000016
112 000050 112265 000017
113
114 000054
115
116 000060 016704 000000G
117 000064 001404
118 000066
119
120 000076
121 000076
122
123 ;+
124 Received message dispatch table
125 ;+
126
127 000100 000076'
128 000102 000120'
129 000104 000552'
130 000106 001022'
131 000110 001176'
132 000112 000076'
133 000114 000120'
134 000116 000076'

 RCVCTL: MOV C.FLG2(R4),N$SNOD(R5)
 MAP C.BUF(R4) ; Map to the buffer
 MOV C.BUF+2(R4),R2 ; Get pointer to message
 MOV (R2)+,R1 ; Get the message flags
 ASR R1 ; Shift to form word offset
 ASR R1 ; ...
 ASR R1 ; ...
 BIC #^C<16>,R1 ; Isolate subtype field

 MOV (R2)+,N$DLA(R5) ; Get destination link address
 MOV (R2)+,N$DLA+1(R5)
 MOV (R2)+,N$SLA(R5) ; Get source link address
 MOV (R2)+,N$SLA+1(R5)

 CALL @RCVTBL(R1) ; Dispatch to processing routine

 MOV $RCCB,R4 ; Did we keep this message?
 BEQ 10$; If EQ, yes
 CAL FLSRES ; Flush away the resources

10$:
 NOOP: RETURN

 ;+
 Received message dispatch table
 ;+
 RCVTBL: .WORD NOOP ; No operation
 .WORD RCVCI ; Connect initiate
 .WORD RCVCC ; Connect confirm
 .WORD RCVDI ; Disconnect initiate
 .WORD RCVDC ; Disconnect confirm
 .WORD NOOP ; Reserved
 .WORD RCVCI ; Retransmitted connect initiate
 .WORD NOOP ; Reserved

```

```

598 .SBTTL Retransmit message
599 +
600 **--RTRANS-Retransmit message
601
602 Retransmit the message on the message retransmission queue (disconnect
603 initiate, connect confirm or retransmitted connect initiate).
604 -
605 Inputs:
606 R3 = Virtual address of LLT
607 R5 = Address of data base descriptor
608
609 Registers modified:
610 R0, R1, R2, R4
611
612 001672 016304 000060 RTRANS::MOV L,RTQ(R3),R4 ; Get message for retransmission
613 001676 001427 BEQ 20$; If EQ, none
614
615 001700 005063 000060 CLR L,RTQ(R3) ; No message waiting now
616 001704 016402 000016 MOV C,BUF+2(R4),R2 ; Compute end address of message
617 001710 066402 000020 ADD C,CNT(R4),R2 ;
618 001714 016464 000024 000016 MOV C,BUF2(R4),C,BUF+2(R4) ;
619 001722 126427 000011 000000 CMPB C,MOD(R4),#NT$CON
620 001730 001006 BNE 10$; If NE, not a connect initiate message
621 001732 MAP C,BUF(R4) ; Map to the data buffer
622 001740 112774 000150 000016 MOVB #MF.CTL!MC.RC,aC,BUF+2(R4)
623
624 001746 10$: CALLE SNSESD ; Send the message
625 001756 116563 000014 000054 20$: MOVB N$DLY(R5),L,TMRD(R3) ; Restart the timer
626 001764 RETURN
627
628 000001 .END

```

```

60 .SBTTL Copy optional data for connect completion
61 ;+
62 ;*-CPYCNC-Copy optional data for connect completion
63 ;
64 ; Copy the optional data from the local optional data buffer to the user
65 ; specified buffer (if supplied).
66 ;
67 ;-
68 ; Inputs:
69 ; R3 = Virtual address of LLT
70 ; R5 = Address of database descriptor
71 ;
72 ; Outputs:
73 ; R1 = # of bytes of optional data copied
74 ; 'C' Clear - No data overrun
75 ; 'C' Set - Data overrun
76 ;
77 ; Registers modified:
78 ; R0, R2
79 ;
80 ;.ENABL LSB
81 CPYCNC: SAVRG <R3> ; Save LLT address
82 CLR -(SP) ; Assume no data overrun
83
84 MOV L.WIND(R3),R2 ; Get address of window block
85 MOV W.TMP(R2),R3 ; Get address of I/O packet
86
87 MOV $OPLNG,R0 ; Get length of optional data in the message
88 CMP R0,1.PRM+16(R3) ; Did user supply enough buffer space?
89 BLOS 10$, ; If LOS, yes
90 MOV 1.PRM+16(R3),R0 ; Use user supplied buffer size
91 INC (SP) ; Set data overrun flag
92
93 10$: MOV R0,-(SP) ; Save byte count
94 BEQ 30$; If EQ, no data move needed
95
96 MOV #$OPDAT,R1 ; Point to optional data
97 MAP 1.PRM+12(R3) ; Map to the user's buffer
98 MOV 1.PRM+14(R3),R2 ; Get pointer to user's buffer
99 20$: MOVB (R1)+(R2)+,R0 ; Copy the optional data
100 SOB R0,20$; ...
101
102 30$: MOV (SP)+,R1 ; Set up count of bytes copied
103 NEG (SP)+ ; Set/Clear 'C' for data overrun indication
104 RESRG <R3> ; Recover LLT address
105 MAPLLT ; and it's mapping
106 40$: RETURN

```

[illegible]

```

330 .SBTTL SLI disconnect link
331 +
332 **SLIDSL-SLI disconnect link
333 -
334 This routine is called to oisconnect a logical link.
335 -
336 Inputs:
337 R4 = Address of CCB
338 C.NSP - Physical address of LLT (LLA)
339 C.BUF1 - Optional disconnect data
340 C.FLG2 - ULA
341 C.STA - Source PDV index
342 R5 = Address of database descriptor
343 -
344 Registers modified:
345 R0, R1, R2, R3
346 -
347 .ENABL LSB
348
349 SLIDSL: MOV C.NSP(R4),R3 ; Get physical address of LLT
350 CALLE ACCLLT ; Gain access to LLT
351
352 TST L.PCTL(R3) ; Are we already disconnecting this link?
353 BNE 30$; If NE, yes ... return no such link
354
355 MOV R4,L.PCTL(R3) ; Save address of disconnect CCB
356 BIT #LT.DIR*400,(R3); Is this a synchronising call?
357 BNE 50$; If NE, yes
358 MOVB #ST$PND,(R3) ; Set state to disconnect pending
359
360 SLIDCT::TST L.TIPI(R3) ; Any transmits in progress?
361 BNE 100$; If NE, yes
362
363 MOV L.PCTL(R3),R4 ; Recover disconnect CCB address
364 CALL MOVDAT ; Move optional data to internal buffer
365 CLR $REASN ; Set reason code to user disconnect
366 CALLR SNDDIS ; Disconnect the link
367

```



823  
824

100\$: SEC  
RETURN

; Indicate unable to queue connect

```

98 .SBTTL Add a new logical link database
99
100 **--ADDLNK--Add a new logical link database
101
102 Allocate the databases needed to add a new logical link.
103
104 -
105 Inputs:
106 R5 = Address of database descriptor
107
108 Outputs:
109 R3 = Virtual address of LLT
110 'C' Clear - Database allocated
111 'C' Set - Failed to allocate a new database
112
113 Registers modified:
114 R0, R1
115
116 ADDLNK::SAVRG <R2,R4
117 MOV #ERRRES,NERRC(R5)
118 MOV N$LVC(R5),R0 ; Get max # of logical links allowed
119 MOV N$LVC+2(R5),R4 ; Get pointer to table
120 TST (R4)+ ; Skip over first entry
121
122 10$: TST (R4)+ ; Is this entry in use?
123 BEQ 20$; If EQ, no
124 SOB R0,10$; Keep looking
125
126 BR 110$; Exit through common code
127
128 20$: MOV #L.LNG,R1 ; Get size of an LLT
129
130 .IF DF N$$MLL
131 CALLX $ALOCX,AUX ; Allocate space
132 BCS 110$; If CS, failure
133
134 MOV R0,N$PLLT(R5) ; Save physical address of LLT
135 MOV R0,-(SP) ; Gain access to the LLT
136 CALL @CEACC ;
137 MOV (SP)+,R3 ; Get virtual address of LLT
138 MOV R3,N$LLT(R5) ; Save virtual address of LLT
139 MOV @KSAR6,N$LLTM(R5)
140
141 MOV R3,R0 ; Copy LLT address
142
143 .IFF
144
145 CALL @ALOCB ; Allocate space
146 BCS 110$; If CS, failure
147
148 MOV R0,N$PLLT(R5) ; Save physical address of LLT
149 MOV R0,N$LLT(R5) ; and virtual address of LLT
150 MOV R0,R3 ; Copy LLT address
151
152 .ENDC
153
154 MOV #L.LNG/2,R1 ; Get length in words

```

```

586 SBTTL Convert CI message to pending connect block
587
588 +
589 **CNVCI-Convert CI message to pending connect block
590
591 Convert the received CI message into an internal pending connect
592 block format.
593
594 Inputs:
595 R2 = Pointer to start of session control information
596 R3 = Virtual address of LLT
597 R4 = Address of CI CCB
598 R5 = Address of database descriptor
599
600 Outputs:
601 R4 = Address of pending connect block CCB
602 'C' Clear - Pending connect block constructed
603 'C' Set - Unable to build pending connect block
604 (resource or format error)
605
606 Registers modified:
607 R0, R1, R2
608
609 CNVCI:: SAVRG <R5,R3> ; Save registers
610 SAVMAP ; and mapping
611 MOV N$SNOD(R5),R0 ; Get source address of incoming message
612 MOV R0,$LADDR ; and store for later comparison
613 BIC #*C<N$SARA>,R0 ; Isolate area address
614 MOV @DECPT,R1 ; Get this node's area address
615 MOV D$LNUM(R1),R1 ; ...
616 BIC #*C<N$SARA>,R1 ; ...
617 CMP R0,R1 ; Are they both the same area?
618 BNE $5 ; If NE, no
619 BIC #N$SARA,$LADDR ; Form alternate address for match
620
621 5$: MOV #E$RES,N$ERRC(R5)
622 CALL @CCBGT ; Allocate a CCB
623 BCC 7$; If CS, allocation failure
624 JMP 130$; BR if allocation failure
625
626 7$: SAVRG <R2> ; Save a register
627 MOV #N.CBL+16.+4,R1 ; Length of pending connect block
628 CALL @ALOCB ; Allocate block
629 RESRG <R2> ; Restore register
630 BCS 110$; If CS, allocation failure
631
632 10$: MOV #E$FMT,N$ERRC(R5)
633 CLR C.FLG2(R4) ; Initialise access control status
634 CLR C.BUF(R4) ; Set up buffer descriptor bias
635 MOV R0,C.BUF2+2(R4) ; Save address for deallocation
636 ASP R1 ; Clear out the pending connect block
637 CLR (R0)+ ; ...
638 SOB R1,10$; ...
639
640 MOV C.BUF2+2(R4),R0 ; Recover block address
641 CMP (R0)+,(R0)+ ; Skip over task name
642 MOV R0,C.BUF+2(R4) ; Set start address of connect block

```

```

1078 .SBTTL Allocate a data buffer for message transmission
1079
1080 ***-GETLDB-Allocate a large data buffer for message transmission
1081 ***-GETSDB-Allocate a small data buffer for message transmission
1082
1083 Allocate a data buffer for message transmission. The first part of
1084 the NSP header will be built into the buffer.
1085
1086 -
1087 Calling sequence:
1088 JSR R1,GETLDB or
1089 JSR R1,GETSDB
1090 .BYTE <Message type code>
1091 .BYTE <Message flags>
1092
1093 Inputs:
1094 R5 = Address of database descriptor
1095
1096 Outputs:
1097 R2 = Pointer to next available byte in the buffer
1098 R4 = Address of the CCB
1099 Mapping will be changed to the data buffer
1100 'C' Clear - Buffer was successfully allocated
1101 'C' Set - Buffer allocation failure
1102
1103 .PSECT
1104 .ENABL LSB
1105
1106 002242 012765 000001 000022 GETLDB::MOV #ERRES,NERRC(R5)
1107 002250 CALL @LDBGT ; Allocate a large data buffer
1108 002254 103441 BCS 100$; If CS, allocation failure
1109
1110 002256 MAP C.BUF(R4) ; Map to the data buffer
1111 002264 000403 BR 10$; Enter common code
1112
1113 002266 GETSDB::CALL @CSBGT ; Allocate a combined CCB and small data buffer
1114 002272 103432 BCS 100$; If CS, allocation failure
1115
1116 10$: MOV #N$SHDR,R2 ; Set up amount of overhead to leave for XPT
1117 ADD C.BUF+2(R4),R2 ; Compute starting address of data area
1118 MOV R2,C.BUF+2(R4) ; Save new buffer address
1119 MOV R2,C.BUF2(R4) ; and for possible retransmission
1120
1121 002314 111164 000002 MOVB (R1),C.RSV(R4) ; Fill in message type code
1122 002320 111164 000011 MOVB (R1),C.MOD(R4) ; and again for possible loopback
1123 002324 016564 000030 000004 MOV N$PLLI(R5),C.NSP(R4)
1124
1125 002332 116122 000001 MOVB 1(R1),(R2)+ ; Fill in the message flags
1126 002336 116522 000016 MOVB N$SLA(R5),(R2)+ ; Fill in destination link address
1127 002342 116522 000017 MOVB N$SLA+1(R5),(R2)+
1128 002346 116522 000020 MOVB N$DLA(R5),(R2)+ ; Fill in source link address
1129 002352 116522 000021 MOVB N$DLA+1(R5),(R2)+
1130
1131 002356 000241 CLC ; Indicate buffer successfully allocated
1132 002360 032101 BIT (R1)+,R1 ; Skip arguments without changing C-bit
1133 002362 000201 RTS R1
1134
1135 .DSABL LSB

```

1538  
 1539  
 1540  
 1541  
 1542  
 1543  
 1544  
 1545  
 1546  
 1547  
 1548  
 1549  
 1550  
 1551  
 1552  
 1553  
 1554  
 1555  
 1556  
 1557  
 1558  
 1559  
 1560

003276 016700 000000G  
 003302 110063 000102  
 003306 001410  
 003310 010302  
 003312 062702 000103  
 003316 012701 000000G  
 003322 112122  
 003324  
 003330

```

.SBTTL Save optional data in LLT
+
**SAVOPT-Save optional data in LLT
-
Save the optional data provided by the user in the LLT.
Inputs:
R3 = Virtual address of LLT
R5 = Address of database descriptor
Registers modified:
R0, R1, R2
SAVOPT::MOV $OPLNG,R0 ; Get length of optional data
MOV R0,L.OPDL(R3) ; and save in LLT
BEQ 100$; If EQ, none
MOV R3,R2 ; Compute address of optional data buffer
ADD #L.OPD,R2
MOV #SOPDAT,R1 ; Point to available optional data
10$: MOVB (R1)+(R2)+ ; Copy the data
SOB R0,10$; ...
100$: RETURN

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE      | REFERENCES                                                      |
|----------|------------|-----------------------------------------------------------------|
| FNDMBX   | 000412 RG  | #26-1063                                                        |
| FR\$BCC  | = 000007   | #6-54                                                           |
| FR\$CCF  | = 000001   | #6-54                                                           |
| FR\$CDF  | = 000002   | #6-54                                                           |
| FR\$DAO  | = 000011   | #6-54                                                           |
| FR\$EXC  | = 000000   | #6-54                                                           |
| FR\$FRM  | = 000010   | #6-54                                                           |
| FR\$FTL  | = 000005   | #6-54                                                           |
| FR\$OPN  | = 000004   | #6-54                                                           |
| FR\$RFD  | = 000006   | #6-54                                                           |
| FR\$SBU  | = 000012   | #6-54                                                           |
| FR\$SHO  | = 000003   | #6-54                                                           |
| FR\$SUBU | = 000013   | #6-54                                                           |
| FR\$UPT  | = 000014   | #6-54                                                           |
| GETLDB   | 002242 RG  | #27-1104                                                        |
| GETSDB   | 002266 RG  | #27-1111                                                        |
| H.LUN    | = ***** GX | *38-1524                                                        |
| IE.ABO   | = ***** GX | 22-895                                                          |
| IOFIN    | = ***** GX | 22-907                                                          |
| IOREDO   | 000452 RG  | #28-1152                                                        |
| I.PRM    | = ***** GX | *22-906 40-1592                                                 |
| I.TCB    | = ***** GX | 40-1595                                                         |
| KILLNK   | 002364 RG  | 12-368 #29-1177                                                 |
| KISAR6   | = ***** GX | *10-259 *11-298 *11-347 17-609 *17-660 *17-720 *19-780          |
|          |            | *19-786 *20-806 *21-850 *21-853 23-929 *23-951 *23-956 *27-1108 |
|          |            | *30-1211 *30-1237 34-1363 *34-1368 *35-1387 *42-1711            |
| KSAR6    | = ***** GX | 8-139                                                           |
| LDBGI    | = ***** GX | 27-1105                                                         |
| LDBRT    | = ***** GX | 36-1465                                                         |
| L.CSTA   | 000037     | *29-1188 41-1636                                                |
| L.CTR    | 000074     | *11-348 20-803                                                  |
| L.DCR    | 000100     | *12-366                                                         |
| L.ILSQ   | 000052     | 35-1406 *35-1408                                                |
| L.LLA    | 000002     | *8-175 20-804 35-1413                                           |
| L.LNG    | 000124     | 8-127 8-154 8-194 35-1421                                       |
| L.LNO    | 000026     | *8-182                                                          |
| L.MASQ   | 000070     | 35-1403                                                         |
| L.NIN    | 000020     | *8-180                                                          |
| L.NXN    | 000016     | *8-179                                                          |
| L.OPD    | 000103     | 10-271 31-1266 39-1555                                          |
| L.OPDL   | 000102     | *10-264 31-1267 *31-1269 *39-1552                               |
| L.REM    | 000006     | *8-178                                                          |
| L.RNO    | 000022     | *8-181                                                          |
| L.RTQ    | 000060     | 35-1389                                                         |
| L.TC     | 000042     | *8-183                                                          |
| L.TIPI   | 000012     | 41-1639                                                         |
| L.USTA   | 000036     | *29-1184 41-1642 42-1699                                        |
| L.WIND   | 000040     | *38-1530                                                        |
| MAPOR!   | 002430 RG  | #30-1211                                                        |
| MO\$SAC  | = 000016   | #6-54                                                           |
| MO\$SPR  | = 000012   | #6-54                                                           |
| MO\$SSV  | = 000014   | #6-54                                                           |

SESTCB11S CREATED BY MACRO ON 28-JUN-85 AT 20:00

PAGE 2 N 15

MACRO CROSS REFERENCE

CREF 04.00

| MACRO NAME | REFERENCES |
|------------|------------|
|            |            |

|       |       |
|-------|-------|
| RESRG | #5-44 |
| SAVRG | #5-44 |

SAVRG #5-44

B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N

